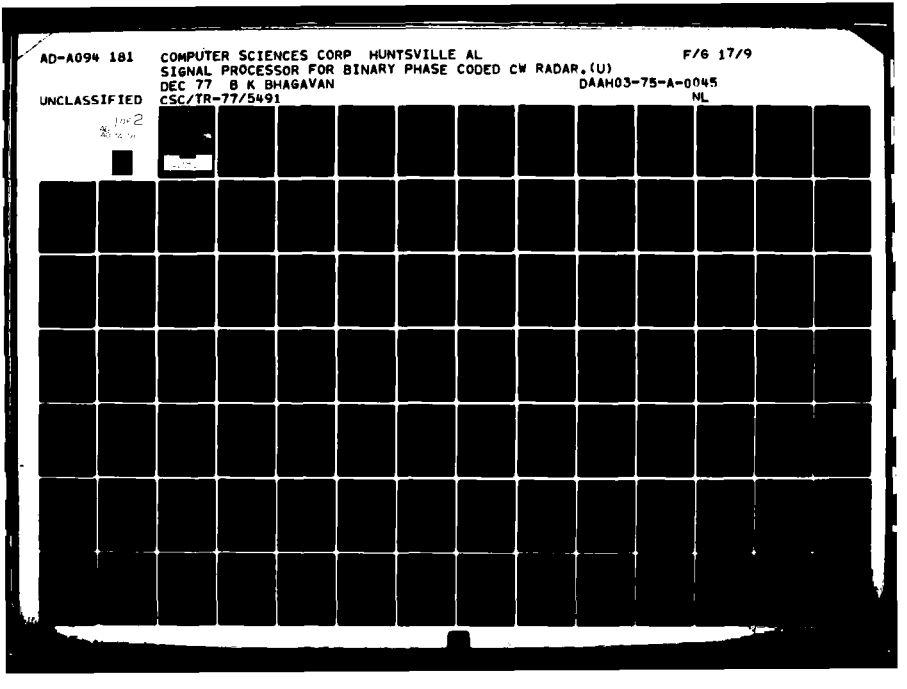
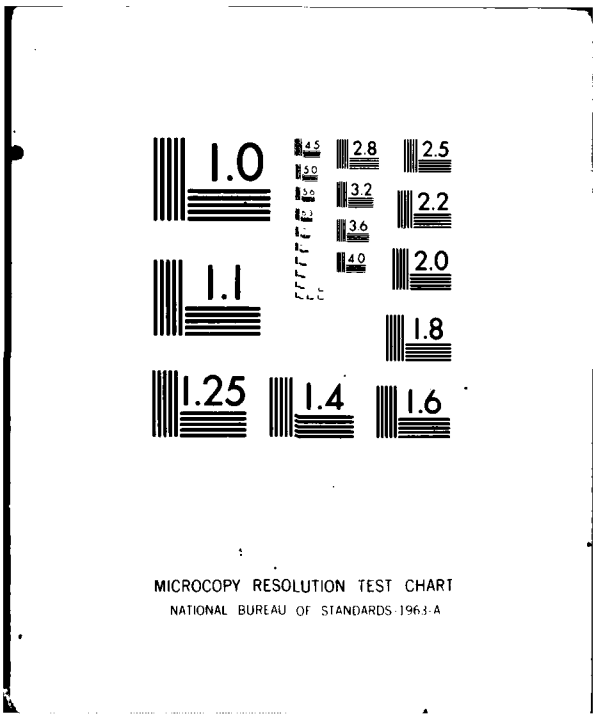
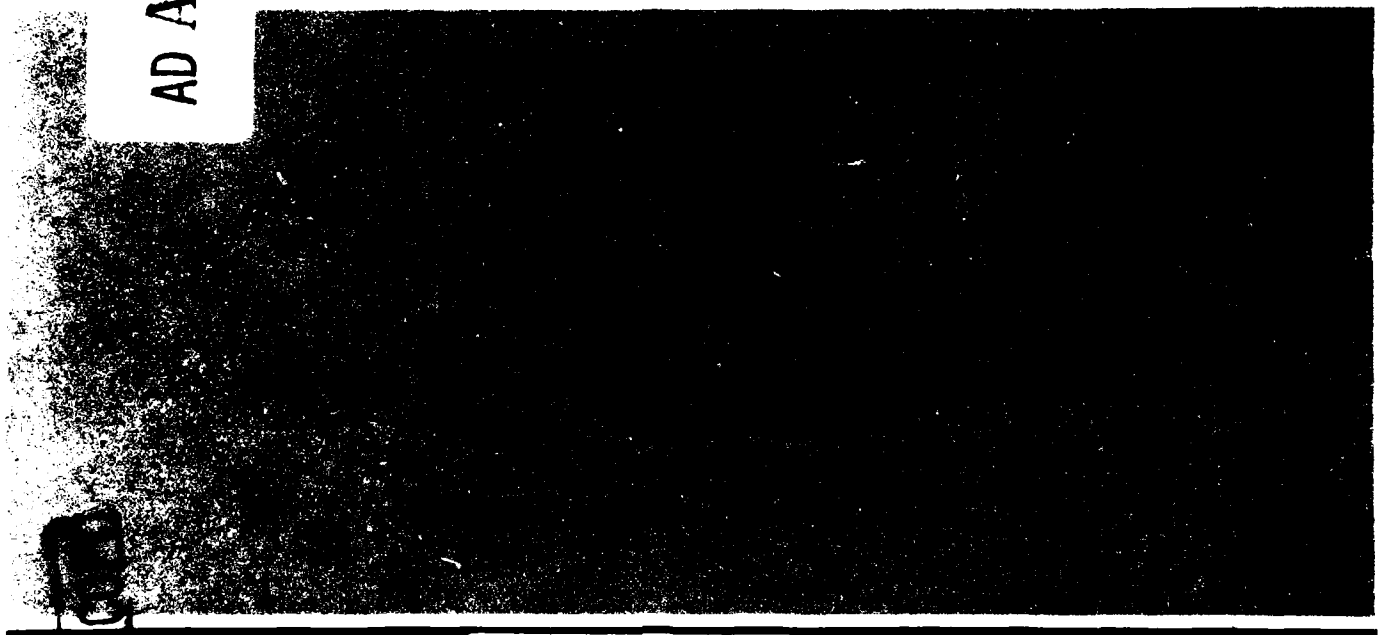
AD-A094 181 COMPUTER SCIENCES CORP HUNTSVILLE AL F/G 17/9
SIGNAL PROCESSOR FOR BINARY PHASE CODED CW RADAR.(U)
DEC 77 B K BHAGAVAN DAAH03-75-A-0045
UNCLASSIFIED CSC/TR-77/5491 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



CSC

COMPUTER SCIENCES CORPORATION

US ARMY MISSILE RESEARCH, DEVELOPMENT AND ENGINEERING LABORATORY
US ARMY MISSILE COMMAND

CLEARANCE OF MATERIAL FOR PUBLIC RELEASE

PART I

Title of Material SIGNAL PROCESSOR FOR BINARY PHASE CODED CW RADAR

Author(s) B K. BHAGAVAN

Organizational Element _____

( ) Technical Report _____
(Report Number)

( ) Open Literature _____
(Title of Journal)

( ) Presentation _____

_____
(Date and Place to be Presented)

This material is based upon unclassified research investigations currently being performed in this Laboratory. There is no objection to open release on grounds of security and accuracy. Applicable security checklists, if any, were used in the review. THIS REPORT HAS BEEN GIVEN AN OPSEC REVIEW AND IS UNCLASSIFIED.

Reviewing Officer

RADAR GROUP LEADER
Title

17 OCT 1980
Date

DRSMI-RER
Organization

PART II

CLEARANCE ACTION

⋈ Subject material has been APPROVED for publication and/or presentation.

( ) Subject material has been DISAPPROVED for publication and/or presentation.

Eunice B. Crutcher
Information Office, AMICOM

22 December 1980
Date

AMSMI-R Form 92, 1 Jan 74 Previous Edition is Obsolete

NEAL B. LAWRENCE
DRSMI-RER B/5400
USMICOM
REDSTONE ARSENAL
ALA 35898
(205) 876-1678

(6) SIGNAL PROCESSOR FOR BINARY PHASE CODED CW RADAR.

(14) CSC/TR-77/5491

(15) CONTRACT DAAH03-75-A-0045

DECEMBER 1977

(1) Technical Rpt.

Developed for:

U. S. Army Missile Research and Development Command
Advanced Sensors Directorate
Radar Technology Branch
Redstone Arsenal, Alabama 35809

(2) B.K. Bhagavan

SUBMITTED BY:

B. K. Bhagavan

APPROVED BY:

Edwin K. Jackson

COMPUTER SCIENCES CORPORATION

515 Sparkman Drive, NW

Huntsville, Alabama 35806

Major Offices and Facilities Throughout the World

409723

COMPUTER SCIENCES CORPORATION
SYSTEMS DIVISION

January 25, 1978
HWS 7383

Dr. Don Burlage
DRDMI-TER
U.S. Army Missile R&D Command
Redstone Arsenal, Alabama 35809

Dear Dr. Burlage:

Enclosed are 8 copies of the CSC report entitled "Signal Processor for Binary Phase Coded CW Radar." This work was performed under Contract DAAH03-75-A-0045, order number CC23.

If you should require additional information, please do not hesitate to call me at 837-7200, ext. 338.

Sincerely,

Ed Jackson

EKJ/bc

Enclosures

Dr. Donald W. Burlage
Technical Monitor
Advanced Sensors Directorate
Radar Technology Branch
U.S. Army, MIRADCOM
Redstone Arsenal, Alabama 35809

PREFACE

This report documents the work performed by Computer Sciences Corporation under Contract DAAH03-75-A-0045. The task reported here is concerned with signal processing for a phase coded CW radar being developed at the Advanced Sensors Directorate of the U.S. Army Missile Research and Development Command, Redstone Arsenal, Alabama. Computer Sciences Corporation wishes to take this opportunity to express its appreciation to those that have contributed directly and indirectly to the work reported herein, in particular, Mr. W. L. Low, Dr. D. W. Burlage and Mr. R. R. Boothe of the Advanced Sensors Directorate.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd.)

LIST OF FIGURES

LIST OF TABLES

SECTION 1 - INTRODUCTION

The need for a short range, low altitude air defense system that is capable of operating at all times and in adverse weather conditions dictates the use of RF devices. Passive RF systems which depend entirely on emissions from attacking aircraft can easily be rendered useless through emission turnoff by the threat aircraft. Consequently, air defense systems have the need for an active RF capability, such as a radar, in order to fulfill their task. However, with the advent of antiradiation missiles (ARM) the susceptibility of active radars to detection by electronic intelligence receivers and attack radar warning receivers has become a major problem. Survivability of such a radar can be increased by designing it so as to deny detection and classification by the ARM receiver.

The detection performance of a radar depends on the signal energy or the average power. Thus for a given detection range, and hence average power, the peak power can be reduced by increasing the pulse width, and continuous wave (CW) transmission requires the minimum peak power. This reduces the output power requirements of the transmitter; and, more importantly, the radar detectability by broadband ARM receivers is reduced. In addition, CW transmission complicates the sorting and designation process by the ARM receiver in that the commonly used discriminants of pulse repetition frequency and pulse width are not available. One other significant characteristic of CW transmission is the inherent multipath problem generated by such signals. This is due to the fact that CW signals have no leading edges while ARM receivers rely heavily on leading edge gating to reduce multipath effects. The multipath problem for the receiver will be more severe if the radar employs techniques such as intentionally illuminating the ground or other multipath scatterers.

A high gain pencil beam antenna should be used to provide a sufficient power density at the target with a low transmitter power level. This low power coupled with low sidelobes on transmit provides an extremely low detectability by a sidelobe ARM. Moreover, since the radar can be detected in the main beam during surveillance, a small beam size confines this detectability to a small portion of the total coverage.

When a battery of radars is operating in the area, frequency agility can be used to complicate the designation process in the ARM receiver. Since mainlobe detection of the radar is possible, beam-to-beam frequency agility makes this detection essentially useless when the radar is not an isolated radar. The possibility of spot jamming by standoff jammers is virtually eliminated by frequency agility.

Since a CW signal without modulation does not provide range resolution, it is necessary to increase the bandwidth of the signal by using a suitable modulation. Selecting the type of waveform, and hence the bandwidth, is a compromise between potentially conflicting performance criteria such as high resolution, unambiguous range and doppler measurements, adequate performance in clutter and ECM environments, low power density for increased covertness, minimum complexity and cost of waveform generation, signal processing, etc. Sawtooth frequency modulation and a binary phase modulated CW carrier are prime candidates for the radar waveform.

The fundamental disadvantages of a phase coded CW radar are inherent to CW radars: antenna isolation and large dynamic range caused by close-in clutter and antenna spillover. This report documents a study of signal processors for a binary phase coded CW radar, where a maximum length shift register generated sequence is used for phase modulation. Following a brief review of phase

coded CW radars and the problems associated with such radars, several analog, digital and hybrid signal processor configurations are proposed. The advantages and drawbacks of each configuration are discussed. In order to evaluate the performance of the proposed processors, digital computer simulations of the input signal and the processors are developed. These simulations are used to demonstrate the feasibility of digital processing in spite of the large dynamic range.

SECTION 2 - PHASE CODED CW RADAR

2.1 INTRODUCTION

Since an unmodulated CW radar cannot resolve targets in range, some form of modulation is required. The ability of the radar system to resolve targets in range depends on the bandwidth of the transmitted modulation; in fact the range resolution is inversely proportional to the bandwidth. Expansion of bandwidth is achieved by modulating the CW carrier with a repetitive modulation waveform. Two such waveforms which are most commonly used are the sawtooth frequency modulation and binary phase coding. In addition to improved range resolution, the enlarged bandwidth produces a lower power density thereby increasing covertness against ARM systems that use narrow-band superhet receivers.

2.2 BINARY PHASE CODED SIGNALS

The simplest form of phase coding is the binary or the phase-reversal code in which the phase of the carrier is shifted by either zero or 180^{o}. The complex representation of such a signal is given by,

$$\psi(t) = u(t)\ e^{j\omega_o t} \tag{2.1}$$

where ω_o is the carrier frequency in radians/sec and u(t) is the modulation waveform. For a binary phased coded signal, the modulation waveform is of the type

$$\left.\begin{aligned} u(t) &= e^{-j\phi_n} \quad \text{for } (n-1)\delta \leq t \leq n\delta \\ \phi_n &= 0 \text{ or } 180^{o} \end{aligned}\right\} \qquad n = 1,2,.... \tag{2.2}$$

where δ is the code clock period.

From (2.2) it is clear that u(t) is a rectangular wave assuming values of +1 or -1, and can be written as

$$u(t) = \sum_{n=1} a_n R_n(t) \tag{2.3}$$

where

$$\begin{aligned} R_n(t) &= 1 \qquad \text{for } (n-1)\delta \leq t \leq n\delta \\ &= 0 \qquad \text{elsewhere} \end{aligned} \tag{2.4}$$

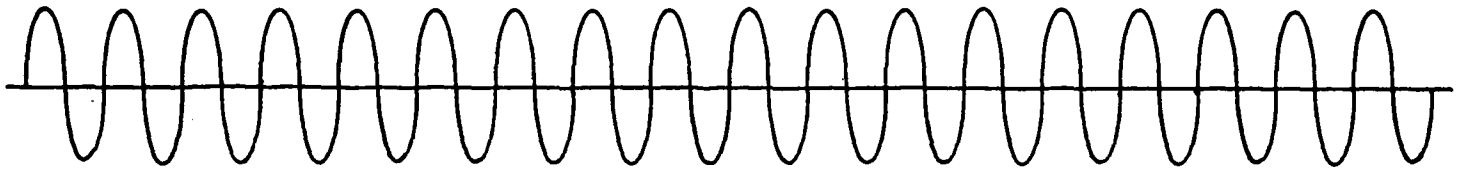
and

$$a_n = \pm 1 \quad . \tag{2.5}$$

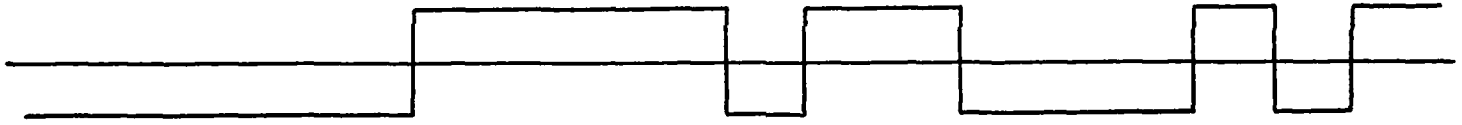
An example of a binary modulating waveform and the corresponding modulated carrier are shown in Figure 2.1. Selection of the modulating waveform is equivalent to choosing the coefficients a_1, a_2, ... so that the signal has the desired range resolution properties.

2.3 PSEUDO-NOISE SEQUENCES

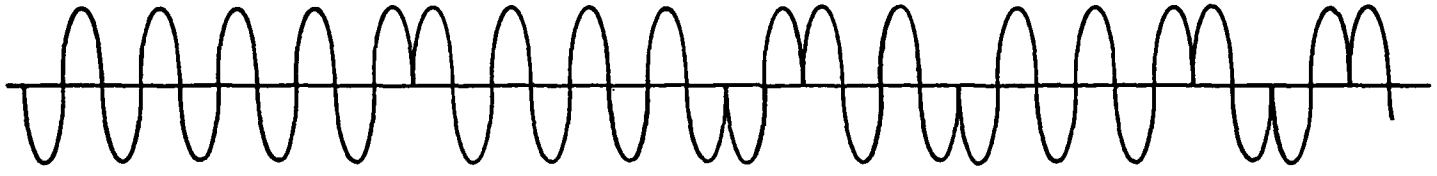
Among the most commonly used binary modulating sequences are the Barker codes and the pseudo-noise (PN) sequences, both of which are periodic and have similar range resolution properties. Barker codes suffer from a major disadvantage in that their existence has been established only for codes with short periods. Pseudo-noise sequences, on the other hand, can be easily generated using linear shift register generators and have been proved to exist for all lengths of the type $N = 2^n-1$. Although pseudo-noise sequences exist for numerous other lengths, only the linear maximum length shift register generated sequences or so-called linear m-sequences are considered in this study.



(a) Unmodulated Carrier



(b) Binary Modulating Waveform



(c) Modulated Carrier

Figure 2.1 Binary Phase Modulation

Linear maximum length sequences can be obtained from the system shown in Figure 2.2. This shift register generator of degree n consists of n storage elements (n stages), a clock pulse generator which generates pulses at intervals of δ, and a feedback logic circuit. When a clock pulse arrives, the contents of each stage are transferred to the next stage, and the content of the last stage is the output. The new state of the first stage is a linear Boolean function of the previous contents of some or all the stages. For each stage, the OFF state is symbolized by 0 and the ON state by 1. The binary output of the feedback logic unit is of the form

$$f(x_1, x_2, \ldots, x_n) = C_1x_1 \oplus C_2x_2 \oplus \ldots \oplus C_nx_n \qquad (2.6)$$

where x_i denotes the state of the i^{th} stage, each constant C_i is either zero or one and the symbol $\oplus$ denotes modulo 2 addition. It can be easily shown that the output sequence of this generator is periodic whose period is at most $N = 2^n-1$. Of the total of 2^n possible feedback combinations, only a few result in sequences whose period is N and such sequences are called linear maximal length sequences. The feedback connections which yield maximal length sequences have been extensively tabulated and a part of it is shown in Table 2.1. The use of this table will be demonstrated with an example. Consider a 6-stage generator, $n = 6$ for which the maximum period is $N = 63$. The table lists three possible feedback connections:

(i) [6,1] which corresponds to

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = x_6 \oplus x_1 \qquad (2.7)$$

(ii) [6,5,2,1] which corresponds to

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = x_6 \oplus x_5 \oplus x_2 \oplus x_1 \qquad (2.8)$$

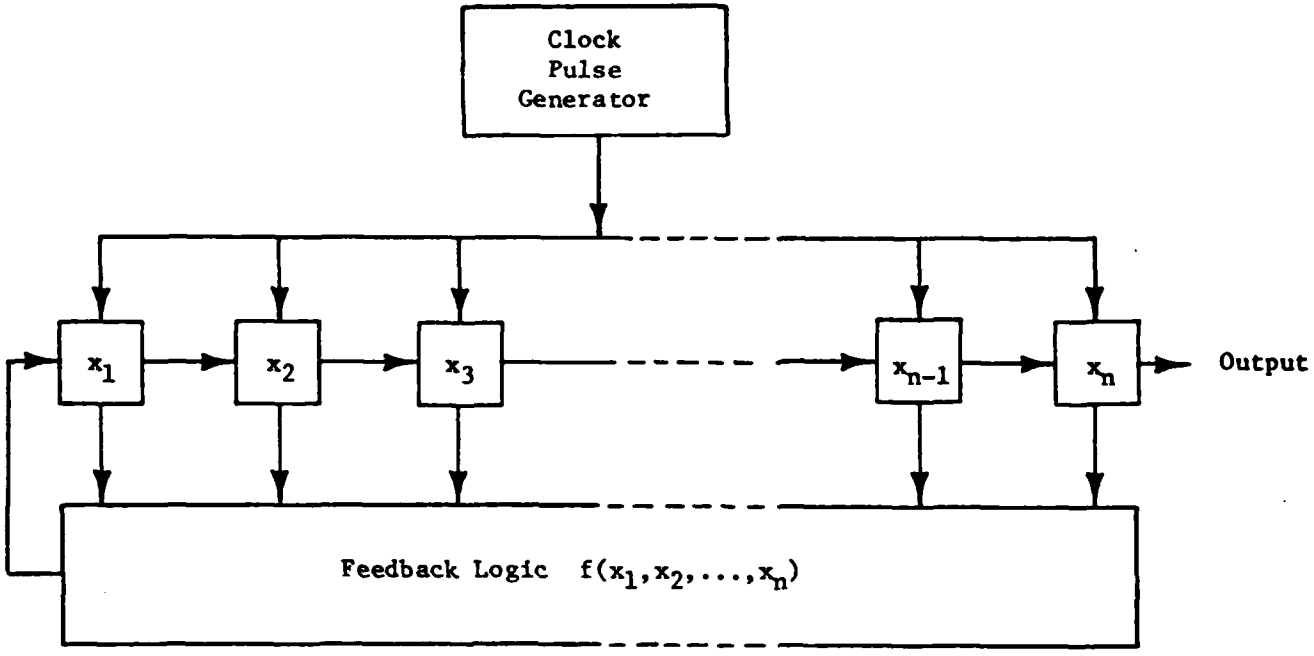


Figure 2.2 Shift Register Generator

Table 2.1 Feedback Connections for Linear m-sequences

Number of stages	Code length	Feedback Connections
2	3	[2,1]
3	7	[3,1]
4	15	[4,1]
5	31	[5,2] [5,4,3,2] [5,4,2,1]
6	63	[6,1] [6,5,2,1] [6,5,3,2]
7	127	[7,1] [7,3] [7,3,2,1] [7,4,3,2] [7,6,4,2] [7,6,3,1] [7,6,5,2] [7,6,5,4,2,1] [7,5,4,3,2,1]
8	255	[8,4,3,2] [8,6,5,3] [8,6,5,2] [8,5,3,1] [8,6,5,1] [8,7,6,1] [8,7,6,5,2,1] [8,6,4,3,2,1]
9	511	[9,4] [9,6,4,3] [9,8,5,4] [9,8,4,1] [9,5,3,2] [9,8,6,5] [9,8,7,2] [9,6,5,4,2,1] [9,7,6,4,3,1] [9,8,7,6,5,3]
10	1023	[10,3] [10,8,3,2] [10,4,3,1] [10,8,5,1] [10,8,5,4] [10,9,4,1] [10,8,4,3] [10,5,3,2] [10,5,2,1] [10,9,4,2]
11	2047	[11,1] [11,8,5,2] [11,7,3,2] [11,5,3,2] [11,10,3,2] [11,6,5,1] [11,5,3,1] [11,9,4,1] [11,8,6,2] [11,9,8,3]
12	4095	[12,6,4,1] [12,9,3,2] [12,11,10,5,2,1] [12,11,6,4,2,1] [12,11,9,7,6,5] [12,11,9,5,3,1] [12,11,9,8,7,4] [12,11,9,7,6,5] [12,9,8,3,2,1] [12,10,9,8,6,2]
13	8191	[13,4,3,1] [13,10,9,7,5,4] [13,11,8,7,4,1] [13,12,8,7,6,5] [13,9,8,7,5,1] [13,12,6,5,4,3] [13,12,11,9,5,3] [13,12,11,5,2,1] [13,12,9,8,4,2] [13,8,7,4,3,2]
14	16383	[14,12,2,1] [14,13,4,2] [14,13,11,9] [14,10,6,1] [14,11,6,1] [14,12,11,1] [14,6,4,2] [14,11,9,6,5,2]

(iii) [6,5,3,2] which corresponds to

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = x_6 \oplus x_5 \oplus x_3 \oplus x_2 . \qquad (2.9)$$

For a given feedback connection, different initial conditions in the shift registers produce the same sequence with different shifts. The initial condition of all zeros produces a sequence of all zeros and is therefore prohibited. The output of the shift register generator is a sequence of zeros and ones which correspond to phase shifts of zero ($a_i = +1$) and 180^o ($a_i = -1$).

The fundamental properties of a linear m-sequence produced by an n-stage generator are:

(i) the sequence is periodic with period $N = 2^n-1$

(ii) in each period, the difference between the number of ones and the number of minus ones is one

(iii) if an m-sequence is multiplied by a shifted version of itself, the resulting sequence is also a shifted version of the original sequence

(iv) the periodic autocorrelation is two-valued, i.e.,

$$\sum_{k=1}^{N} a_i \, a_{i+k} = \begin{cases} N & \text{if } i=0, \pm N, \pm 2N, \ldots \\ -1 & \text{otherwise .} \end{cases} \qquad (2.10)$$

2.4 PERFORMANCE OF PHASE CODED CW RADARS

The ability of the radar system to resolve targets in range depends on the autocorrelation of the transmitted modulation. The ideal autocorrelation has a peak at zero delay and is zero elsewhere. Consider a PN sequence of period N and clock period δ. The code repetition interval is $\Delta = N\delta$. The normalized autocorrelation of such a sequence can be shown to be periodic with period Δ and

has a shape as shown in Figure 2.3. Due to the periodicity of the autocorrelation, the power spectrum is a line spectrum and it can be shown that its envelope is given by

$$\left|U(f)\right|^2 = \Delta \left(\frac{N+1}{N^2}\right) \; sa^2 \; (\pi f/B) \qquad (2.11)$$

where $B = 1/\delta$, except at DC where the value is Δ/N^2. The magnitude spectrum of the waveform (square root of the power spectrum) is sketched in Figure 2.4. From the magnitude spectrum, it can be seen that the bandwidth of the signal is approximately equal to B, the basic clock rate.

While the use of a very wide bandwidth signal offers some improved covertness against ARM receivers which use narrowband superhet receivers, it also has several disadvantages in regard to the radar performance. The very high range resolution caused by the increased bandwidth poses such problems as range-gate flythrough and the need for more receiver channels to cover all the desired range cells in the required time. A wide bandwidth signal is more likely to encounter spot jammers spaced randomly across the operational band. The number of frequency bands available for frequency agility is sharply reduced by the use of wide bandwidth signals. Finally, the large bandwidth will have an undesirable cost impact on the RF portion of the radar and, in particular, the low sidelobe antenna. Although very large bandwidths are not desirable, the bandwidth must be sufficiently large so as to provide good range resolution and improved performance in heavy clutter.

The substantial benefits of a modulated CW radar are somewhat tempered by the inherent drawbacks of a CW radar. Chief among these are the multipath effects, antenna spillover and large receiver dynamic range requirements caused by close-in clutter. Antenna spillover can be minimized by properly shielding the receive antenna from the transmit antenna. The task of designing a signal processor, especially a digital processor, is made considerably more difficult by the large dynamic range requirement.

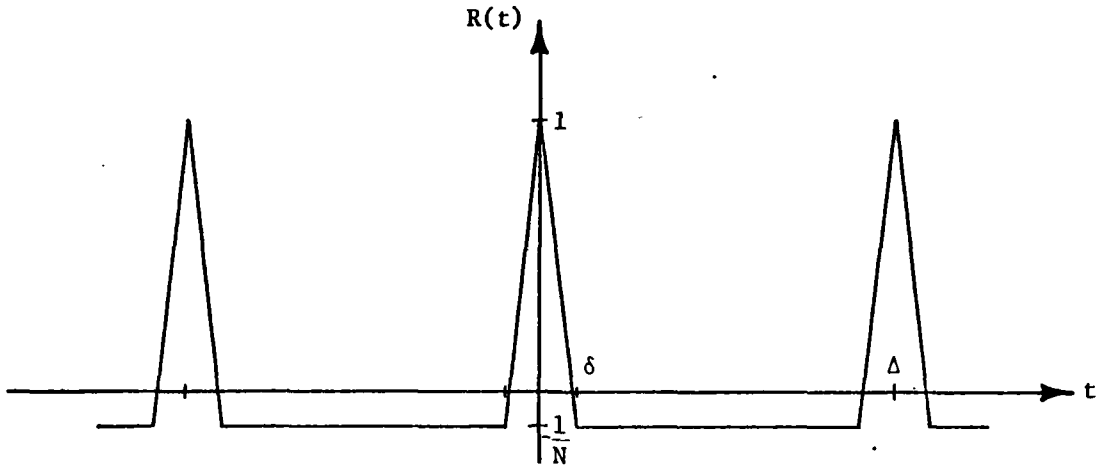


Figure 2.3 Autocorrelation of PN Sequence

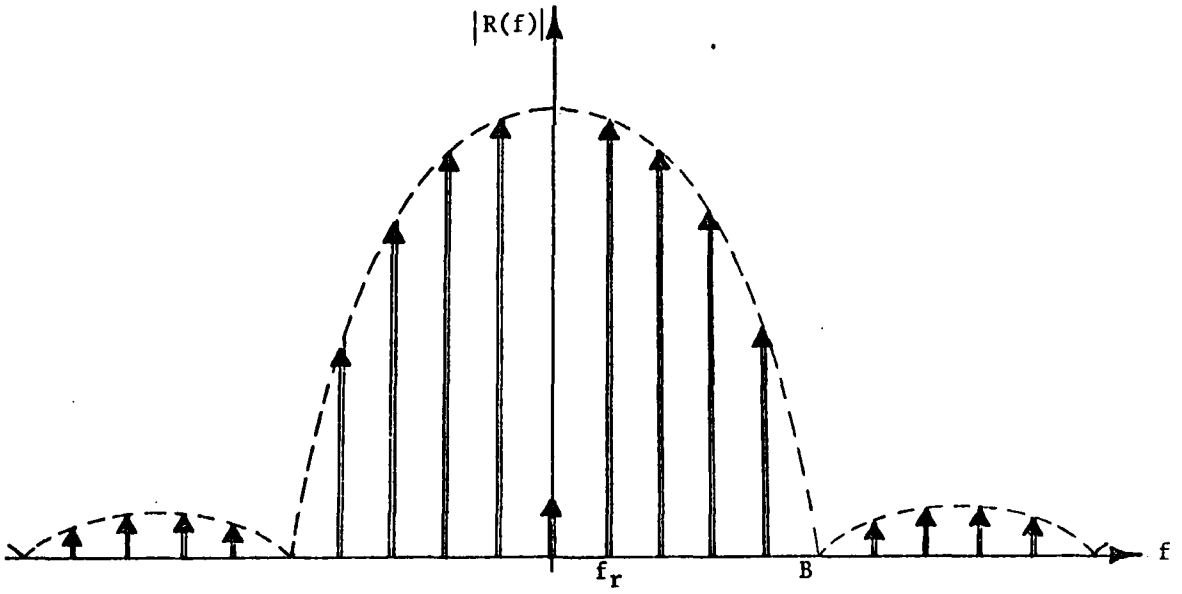


Figure 2.4 Magnitude Spectrum of PN Sequence

2.5 TYPICAL PARAMETER VALUES

The following is a list of symbols and their typical values that will be used throughout the report to demonstrate the operation of the signal processors.

Carrier Frequency:	f_o	= 10 GHz
Wavelength:	λ	= 0.03 Meters
Intermediate Frequency:	f_i	= 30 MHz
Clock Rate:	B	= 5 MHz
Range Resolution:	ΔR	= 30 Meters
Code Length:	N	= 63
No. of Shift Register Stages:	n	= 6
Unambiguous Range:	R_{unamb}	= 1890 Meters
Code Repetition Period:	Δ	= 12.6 μ sec
Code Repetition Frequency:	f_r	$\cong$ 80 KHz
Look Time:	T	= 2 m sec
No. of Code Periods in One Look:	K	= 158
Maximum Range Rate of Interest:	V_{max}	= 360 meters/sec
Maximum Doppler:	f_d	= 24 KHz

Note that due to the relatively small unambiguous range, additional means such as varying the pulse repetition rate must be employed to resolve range ambiguities. The existence of the dynamic range problem can be easily demonstrated by considering a 1 m^2 target at 15 KM range and a 10^4 m^2 fixed clutter at a 2 KM range. For such a situation the signal-to-clutter ratio is

$$SCR = \left(\frac{1}{15^4}\right) \Big/ \left(\frac{10^4}{2^4}\right)$$

$$= -75 \text{ dB}$$

In addition to antenna spillover and noise, various types of clutter must be considered in the design of a signal processor. These include in-range clutter (in the same range cell as the target), out-of-range distributed clutter, fixed clutter such as buildings and weather clutter.

SECTION 3 - THE ANALOG PROCESSOR

3.1 INTRODUCTION

The signal processor described in this chapter is a simple analog processor and is presented to demonstrate the principles of signal processing for phase coded CW radars. Some of the problems associated with such a system are also elaborated. Typical parameter values set forth in the previous chapter are used to explain the operation of the processor. It must be noted that this processor is similar to the one being used to demonstrate the feasibility of the phase coded CW radar concept.

3.2 DESCRIPTION OF THE PROCESSOR

A simplified schematic of the processor is shown in Figure 3.1. The received RF signal is mixed down to a convenient IF frequency f_i and amplified. This signal is then sent into 63 parallel range channels through a power divider. Each channel contains a decoder or a code demodulator where the incoming signal is multiplied by the binary code but with a different shift in each channel. The target return may be written in the form,

$$g_T(t) = p(t-\tau) \cos [2\pi t(f_i+f_d) + \phi] \qquad (3.1)$$

where $p(t)$ is the periodic code,

f_d is the target doppler and

τ is the range delay.

The in-range clutter has a similar form,

$$g_c(t) = a(t)\, p(t-\tau) \cos (2\pi t f_i + \phi_c) \qquad (3.2)$$

where $a(t)$ is the low frequency amplitude variation.

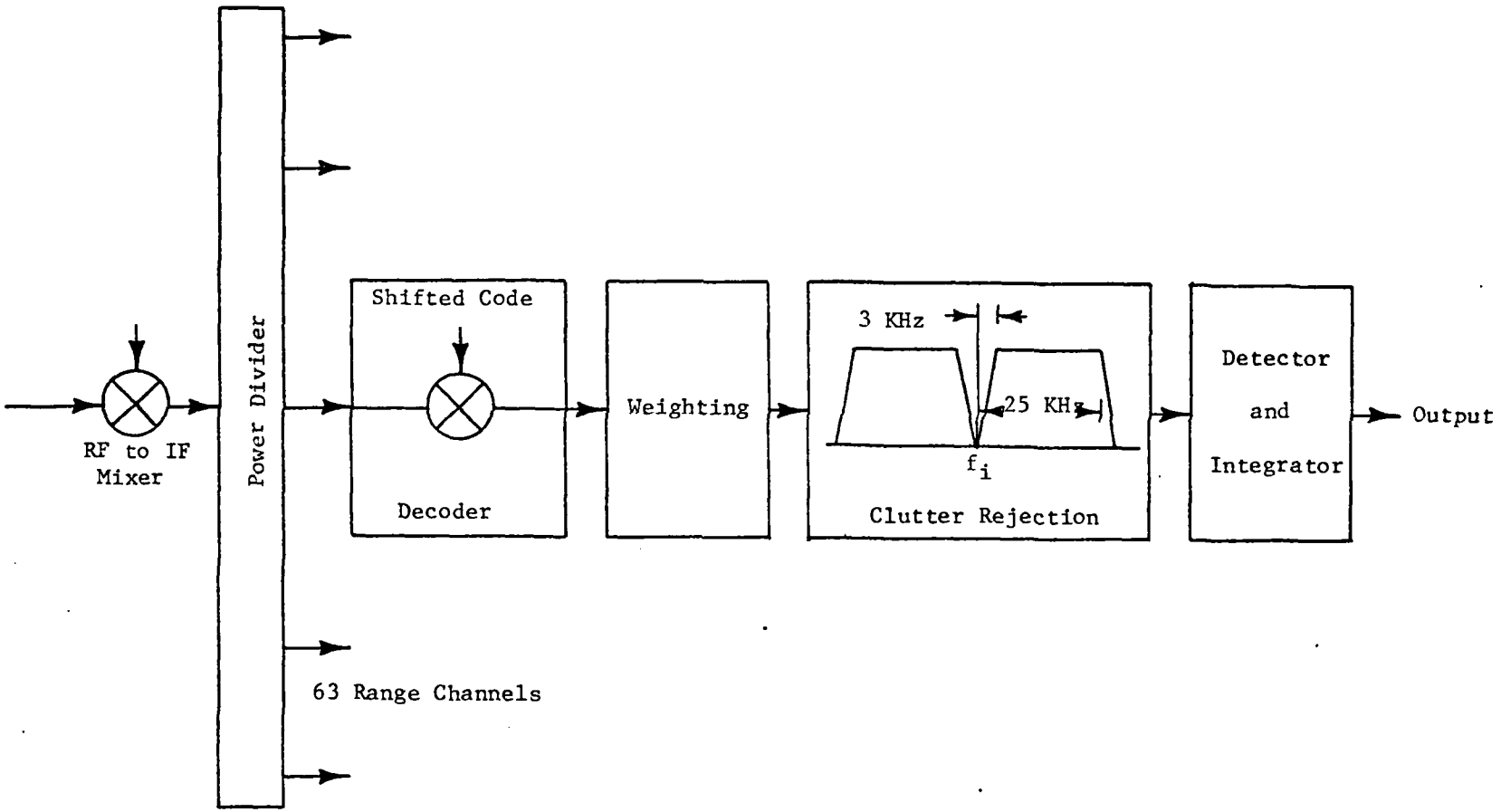


Figure 3.1 The Analog Processor

The out-of-range distributed clutter is the sum of the returns from several range cells, each of the form

$$g_d(t) = b(t)\, p(t-\tau_d) \cos(2\pi t f_i + \phi_d) \qquad (3.3)$$

where $\tau_d \neq \tau$.

One of the range channels contains a decoder which has the same delay as the target. This channel will be called the in-range channel and the other channels the out-of-range channels. The decoder output due to the target in the in-range channel is,

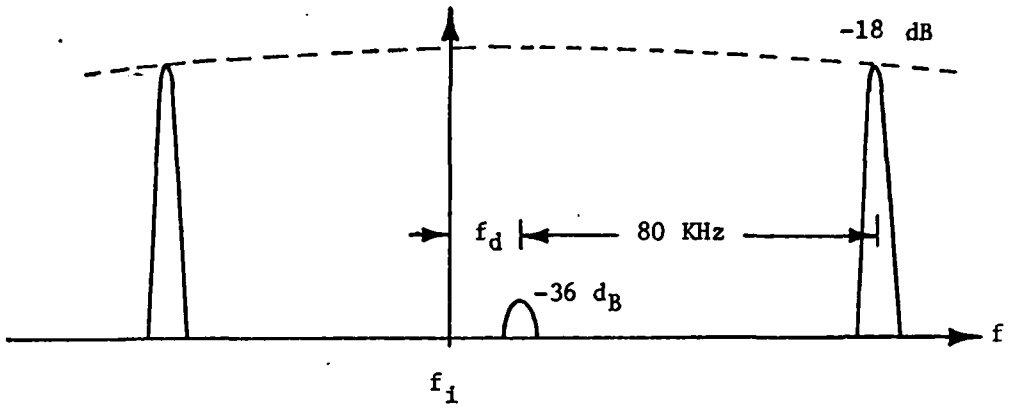
$$\begin{aligned} r_T(t) &= g_T(t)\, p(t-\tau) \\ &= p(t-\tau)\, p(t-\tau) \cos[2\pi t (f_i+f_d) + \phi] \\ &= \cos[2\pi t(f_i+f_d) + \phi] \end{aligned} \qquad (3.4)$$

since $p(t-\tau)$ can only assume values +1 and -1.

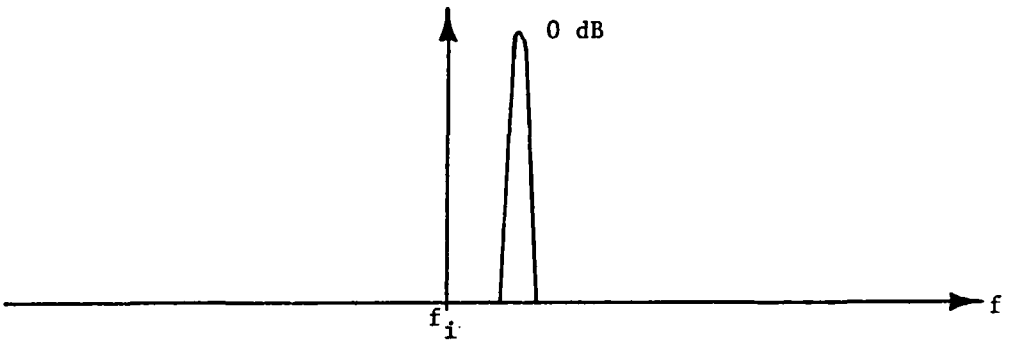
The decoder output in an out-of-range channel due to the target is,

$$\begin{aligned} s_T(t) &= g_T(t)\, p(t-\tau_1) \quad , \tau_1 \neq \tau \\ &= p(t-\tau)\, p(t-\tau_1) \cos[2\pi t(f_i+f_d) + \phi] \\ &= p(t-\tau_2) \cos[2\pi t(f_i+f_d) + \phi] \end{aligned} \qquad (3.5)$$

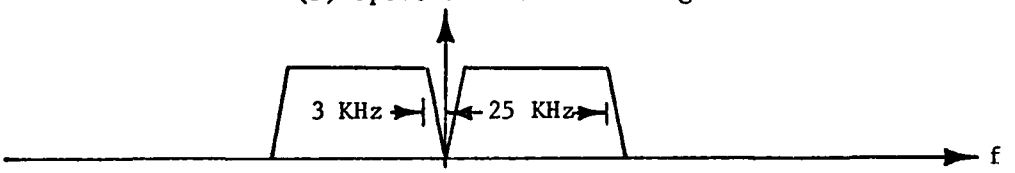
where $\tau_2 \neq 0$ and use is made of the fact that a PN sequence multiplied by its shifted version yields the same sequence with a different shift. The output of the decoder is filtered by a bandpass filter with a clutter notch centered at f_i. Let the outputs of the bandpass filter be $u_T(t)$ and $v_T(t)$ when the inputs are $r_T(t)$ and $s_T(t)$ respectively. The spectra of the signals in the in-range and out-of-range channels due to the target return are shown in Figures 3.2 and 3.3.



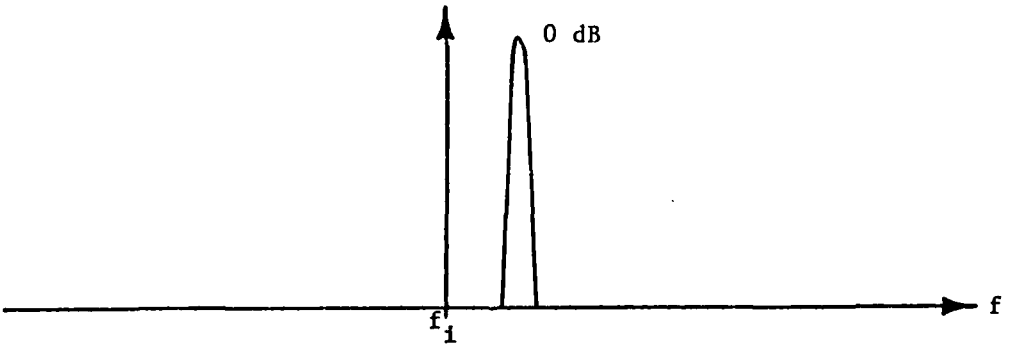
(a) Input Signal Spectrum



(b) Spectrum After Decoding

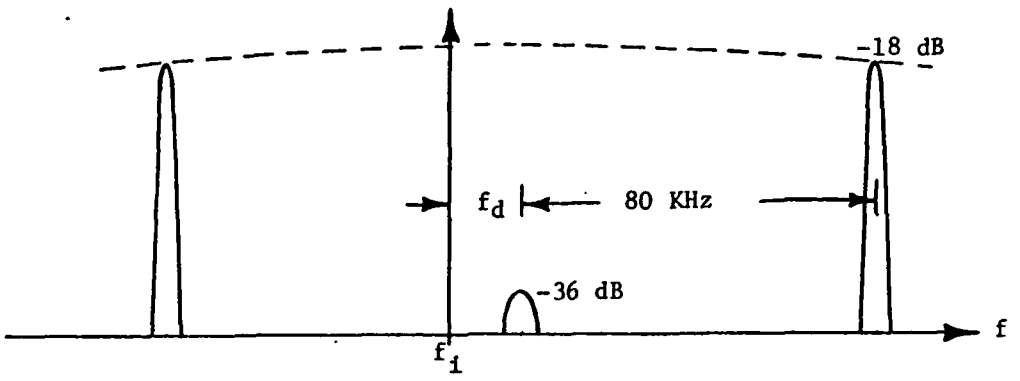


(c) Filter Characteristics

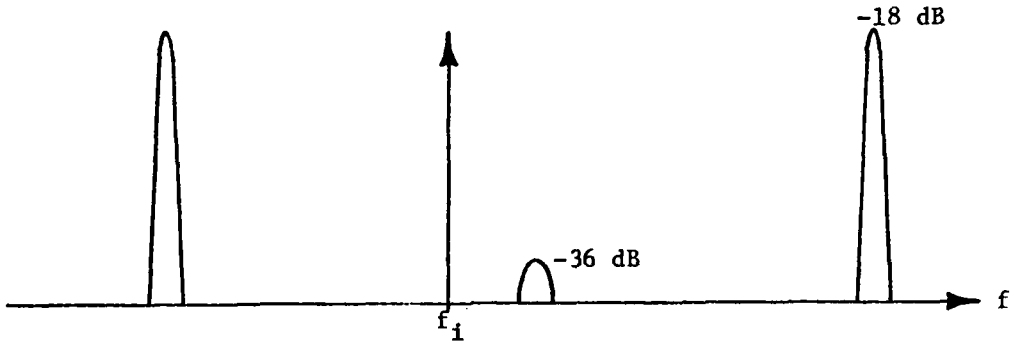


(d) Output Signal Spectrum

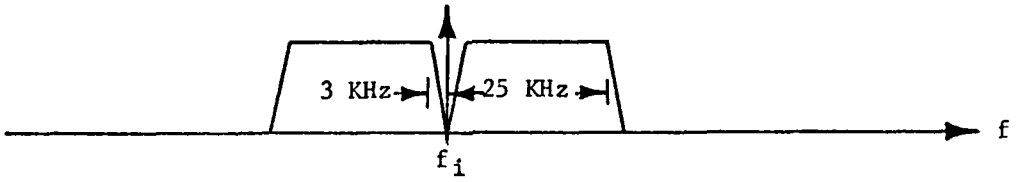
Figure 3.2 Target Return: In-Range Channel



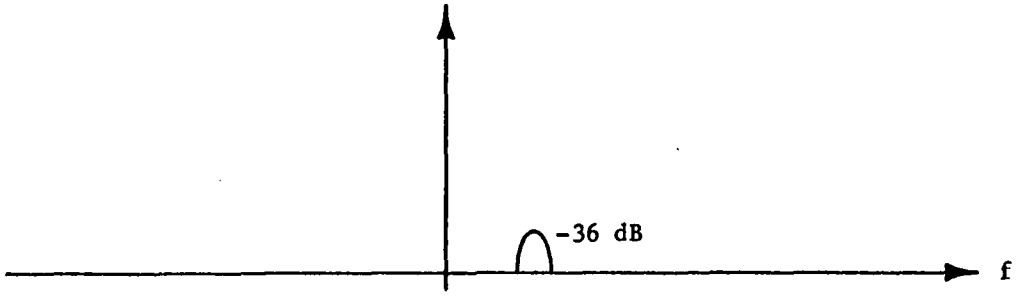
(a) Input Signal Spectrum



(b) Spectrum After Decoding



(c) Filter Characteristics



(d) Output Signal Spectrum

Figure 3.3 Target Return: Out-of-Range Channel

Note that the line spectrum of Figure 2.4 has now been replaced by a spectrum of non-zero width. This is due to the finite look time of 2 msec which corresponds to a spectral width of 500 Hz.

The decoder outputs due to the in-range and out-of-range clutter can be written as,

$$r_c(t) = a(t) \cos(2\pi t f_i + \phi_c) \tag{3.6}$$

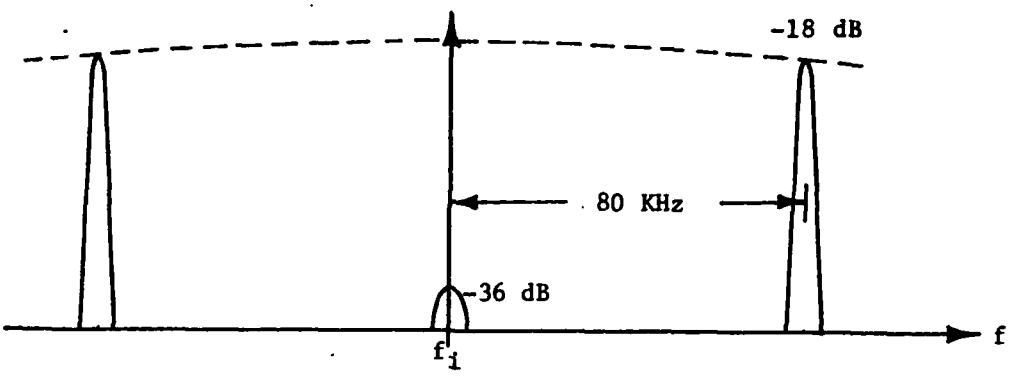
$$r_d(t) = b(t)\, p(t-\tau_3) \cos(2\pi t f_i + \phi_d) \tag{3.7}$$

where $\tau_3 \neq 0$. Let the corresponding outputs of the bandpass filter be $u_c(t)$ and $u_d(t)$. The spectra of the clutter signals in the in-range and out-of-range channels are shown in Figures 3.4 and 3.5.

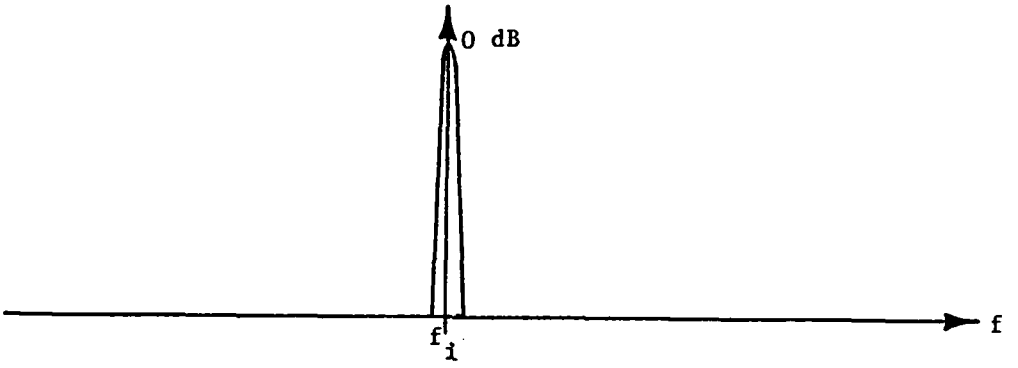
The output of the bandpass filter is passed through a detector and then integrated. The output of the integrator is presented to the thresholding device for detection.

3.3 ANALYSIS OF THE PROCESSOR

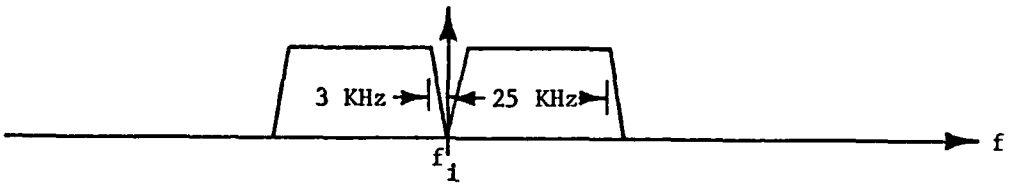
Several observations can be made from an examination of the developments of the previous section. The response due to a target in an out-of-range channel is nearly 36 dB below that in the in-range channel. Therefore, the effect of range sidelobes is negligible. Improvement in signal-to-noise ratio is derived by reducing the bandwidth from the original 5 MHz to 50 KHz at the bandpass filter output. This represents a 20 dB gain in the signal-to-noise ratio. Both in-range and out-of-range clutter can be attenuated by the notch in the bandpass filter. The notch is 6 KHz wide corresponding to a doppler frequency range of -45 to +45 meters/sec. Targets with radial velocities of less than 45 meters/sec cannot be detected.



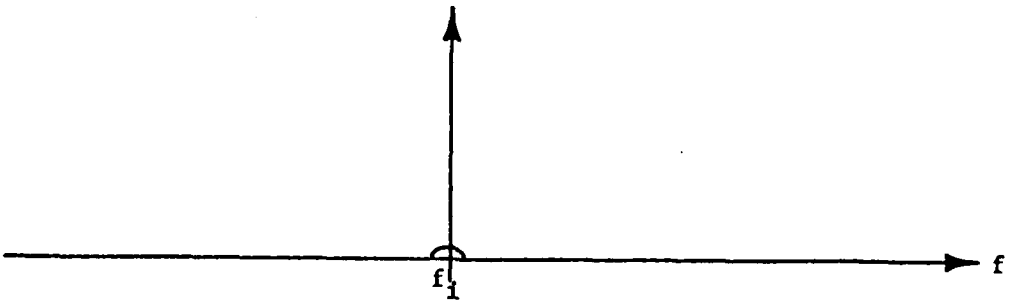
(a) Input Clutter Spectrum



(b) Spectrum After Decoding

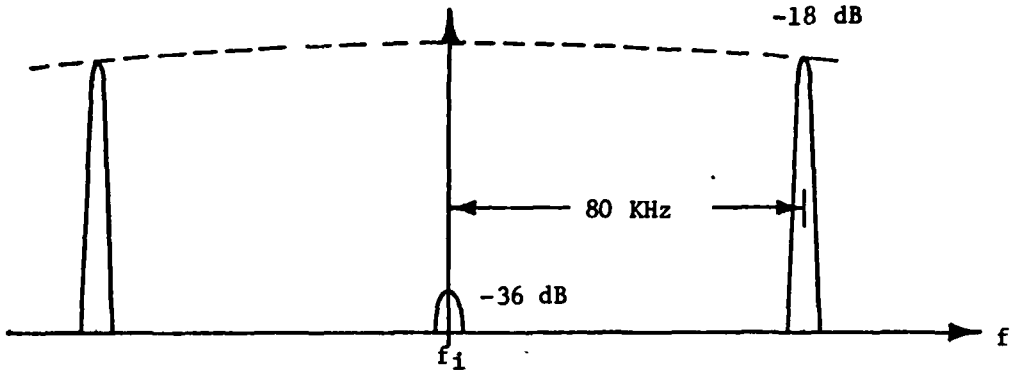


(c) Filter Characteristics

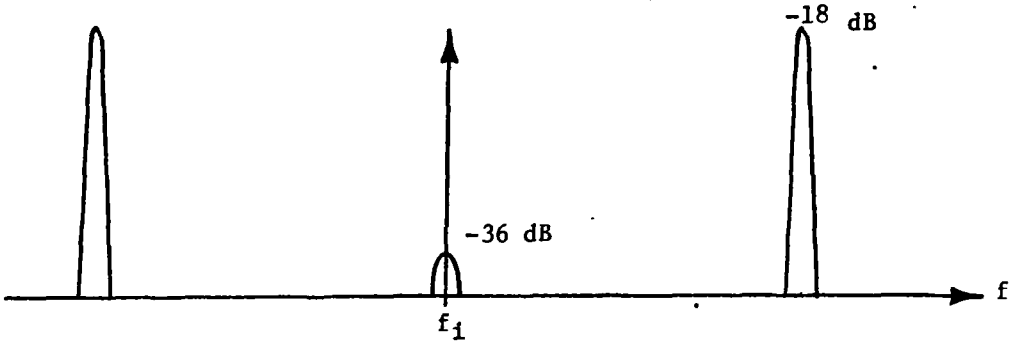


(d) Output Clutter Spectrum

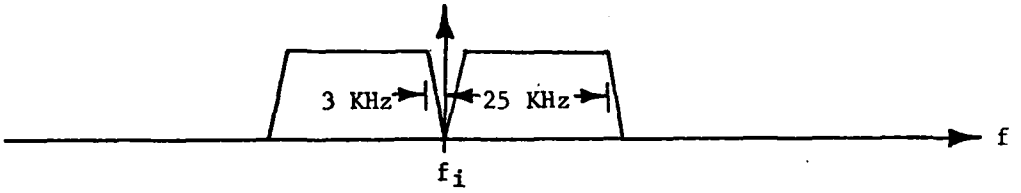
Figure 3.4 In-Range Clutter



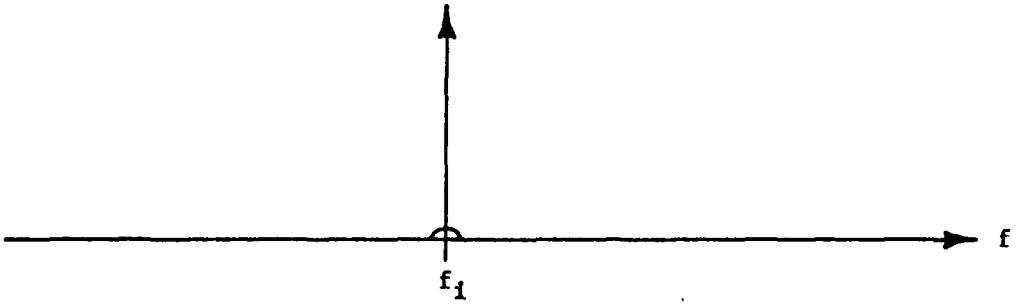
(a) Input Clutter Spectrum



(b) Spectrum After Decoding



(c) Filter Characteristics



(d) Output Clutter Spectrum

Figure 3.5 Out-of-Range Clutter

The dual requirements of adequate clutter suppression and low target return attenuation dictate the need for steep transitions at the notch. This, however, results in a long settling time for inputs in the stop band. Thus, the transient response due to the large clutter inputs persist for a long period of time, thereby severely degrading the clutter cancellation performance of the processor. The problem of long settling time may be alleviated to some extent by using a suitable weighting scheme prior to filtering. Even with the weighting, it will be necessary to wait for the output to settle before starting integration. The weighting and the loss of integration time result in a reduced gain in signal-to-noise ratio. One other problem associated with such a processor is the reinitialization of the filter. This is because the filter must be cleared before processing begins in the next look period. The problem of clearing is not straightforward in the crystal filters needed for the IF bandpass filtering.

3.4 CONCLUSIONS

A simple analog signal processor was presented which uses several range channels, IF notch filtering for clutter reduction, and non-coherent integration. The principle of operation of the processor was explained for the case of a target in clutter. Eventhough the processor is effective, it suffers from the disadvantage of slowly decaying transient response and the problem of clearing the filter prior to the next look period.

SECTION 4 - DIGITAL PROCESSING

4.1 INTRODUCTION

As explained in the last chapter, one of the major drawbacks of the analog processor is the long settling time in the clutter rejection filter. An alternative would be to employ digital processing where it is much easier to design filters with the desired characteristics. For example, finite impulse response or transversal filters can be designed with very short settling times. Simple examples of such filters are the commonly used two and three pulse cancellers. Other advantages of digital processing over analog processing include the wide flexibility offered by digital processing and the possibility of employing coherent integration through the use of fast Fourier transform (FFT) processors. However, digital processing has one significant disadvantage compared to analog processing, it being the additional errors introduced due to quantization. This chapter discusses some of the factors that must be considered during analog-to-digital conversion or digitization.

4.2 ANALOG-TO-DIGITAL CONVERSION

Analog-to-digital (A/D) conversion consists of two steps: sampling and quantization. Sampling is the process by which the analog signal is converted into a discrete time signal. The time interval between two successive samples, usually constant, is called the sampling interval and its reciprocal the sampling rate or sampling frequency. While the sampled signal exists only at discrete instants of time, the amplitude can assume a continuous range of values. Quantization is the process where the amplitudes are approximated by a discrete set of values. This is necessary because in digital representation, the amplitude must be represented by a finite number of binary bits (finite word length). For example

if the word length is 10 bits, then there are only $2^{10} = 1024$ discrete levels of amplitude that can be represented. The approximation of the amplitude by a discrete set of values introduces an error known as the quantization error or quantization noise.

In accordance with the low-pass sampling theorem, the sampling rate must be at least twice the highest frequency at which the signal has a significant component. It therefore becomes obvious that A/D conversion cannot be achieved at RF or at IF because of the unrealistic sampling rate requirements. The signal must therefore be shifted down to video prior to A/D conversion. Since the signals of interest are narrowband signals, the signal can be brought down to baseband, without loss of information, by using two video channels. The in-phase and quadrature signals can be obtained from the IF signal by simply mixing it with sine waves at IF but with a 90^{o} phase shift, as shown in Figure 4.1. The in-phase and quadarture signals can be considered to be the real and imaginary parts of a complex video signal. The spectrum of this signal is similar to the one shown in Figure 2.4 except for a shift due to the doppler frequency. For the set of typical parameters being considered in this study, the bandwidth of this signal is about 5 MHz. Thus, the complex video signal can be sampled at 5 MHz without appreciable loss of information.

The process of quantization can be explained with a simple example. Consider an input signal whose range is -1 to 1 volt and a word length of 3 bits. Assuming linear quantization, the input-output relationship of the digitizer is as shown in Figure 4.2, where the quantity q is called the quantization interval. It is evident from the figure that if z is the middle of one of the steps, then any input in the range z-q/2 to z+q/2 yields the same output. The effect of quantization can therefore be looked upon as the introduction of an error whose value varies between -q/2 and q/2. To facilitate the analysis, the error is

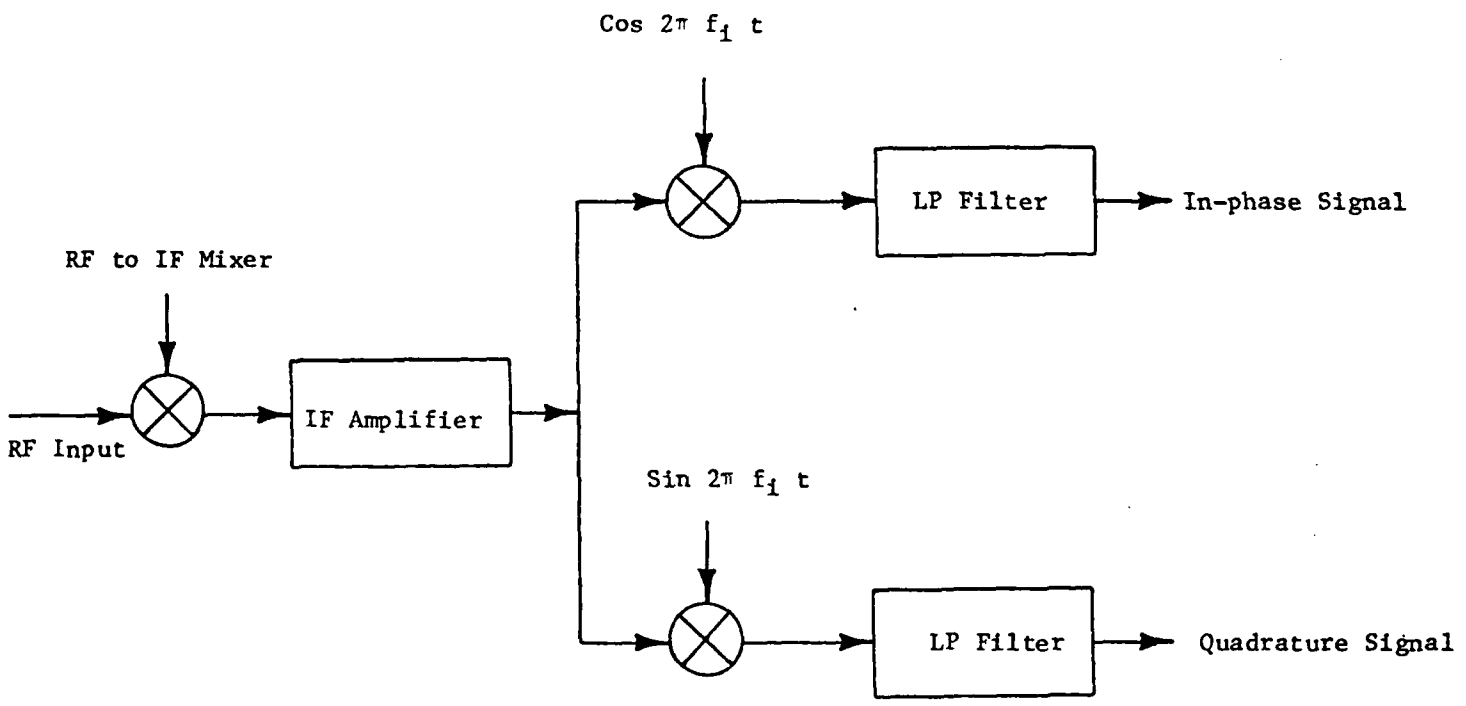


Figure 4.1 Extraction of In-phase and Quadrature Signals

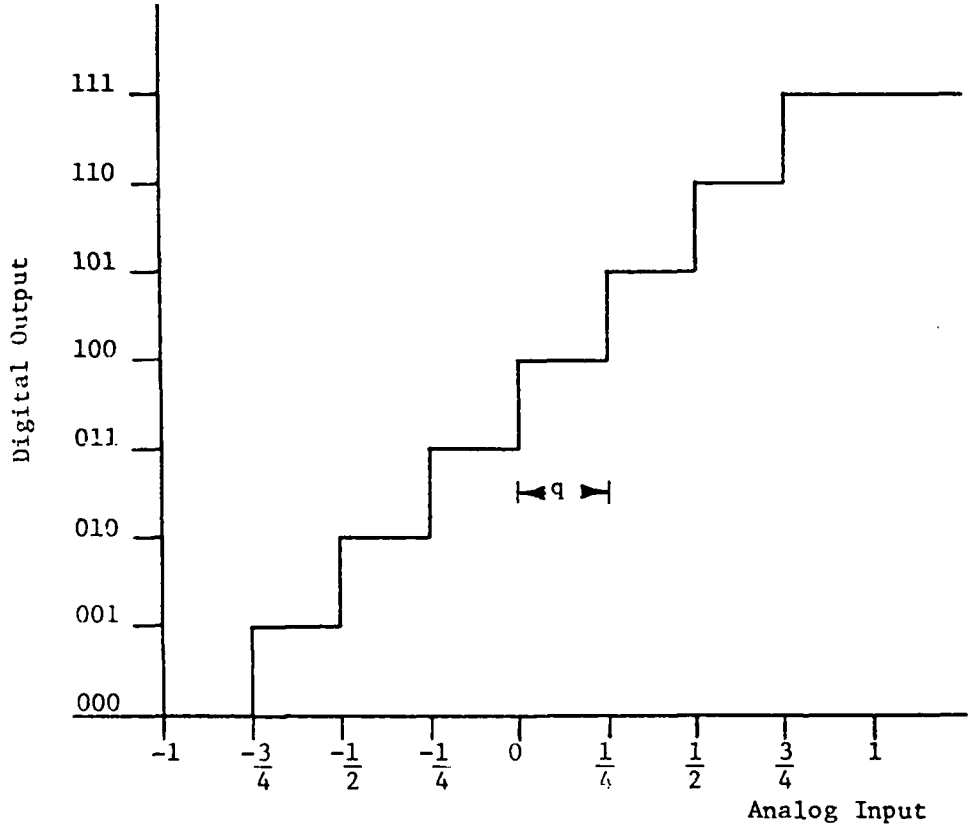


Figure 4.2 A/D Converter Characteristics

considered to be a random variable which is uniformly distributed between -q/2 and q/2. It can be easily shown that such a random variable has zero mean and a variance given by

$$\sigma_q^2 = q^2/12 \quad . \tag{4.1}$$

If the range of the input is -V to +V volts, and if the word length is m-bits, then it can be easily shown that

$$q = 2V/2^m \quad . \tag{4.2}$$

In particular, if V = 1 volt, then

$$q = 2^{-(m-1)} \quad . \tag{4.3}$$

The dynamic range of the A/D converter is defined as the ratio of the largest signal power to the quantization error power. Assuming a sine wave input, the largest signal that can be converted without saturation has a magnitude of V and a power of $V^2/2$. Thus, the dynamic range in dB is,

$$\begin{aligned} R_{dB} &\triangleq 10 \log_{10} \frac{V^2/2}{q^2/12} \\ &= 6.02m + 1.76 \\ &\cong 6m \text{ dB} \end{aligned} \tag{4.4}$$

To gain insight into the dynamic range problem posed by the CW radar, consider a signal-to-clutter ratio of -70 dB. If the signal power is 1, then the clutter variance is

$$\sigma_c^2 = 10^7 \tag{4.5}$$

and clutter standard deviation is

$$\sigma_c = 3.16 \times 10^3 \quad . \tag{4.6}$$

If the A/D converter input range is adjusted so that a $3\sigma_c$ signal is not saturated, then

$$V = 3\sigma_c$$

$$= 9.5 \times 10^3 \quad . \tag{4.7}$$

With a 13-bit A/D converter, the quantization interval is

$$q = 2V/_2 13$$

$$= 2.3 \text{ volts} \quad . \tag{4.8}$$

Thus, the total excursion of the target return is less than q, and if the target return were not corrupted by clutter and noise, it is possible that the output of the A/D converter is a constant, and that the target return will be completely lost. One way to avoid this problem is to use a larger word length, but the word length is limited by the 5 MHz sampling rate requirement.

4.3 PRELIMINARY EXPERIMENTS

Preliminary simulation experiments were performed to examine the feasibility of detecting a sine wave in clutter when the signal amplitude is considerably smaller than the quantization interval. The clutter signal included both DC clutter and random AC clutter with Gaussian distribution and Gaussian spectrum. A sampling rate of 5 MHz was used, and a matched filter matched to the sine wave for a period of 2 msec was used as the detector. It was found that the sine wave could be detected in 80 dB clutter with a word length as small as 10 bits. Based on these results, it was inferred that digital processing is feasible for the CW radar despite the stringent dynamic range requirements posed by the wide disparity between the magnitudes of the clutter return and the target return.

4.4 OTHER CONSIDERATIONS

In addition to the quantization errors introduced during A/D conversion, roundoff errors are introduced during every arithmetic operation. The magnitudes of these errors depend on the word length, the type of number representation (one's complement, two's complement, etc.) and the type of arithmetic employed (fixed point, floating point, etc.). In the case of a complicated processor, such as the FFT processor, these errors are substantial and must be taken into account in the design. Another effect of the use of finite word length is the change in the characteristics of a filter due to truncation of the coefficients. The pole-zero structure and hence the frequency response are altered and in some extreme cases the filter may become unstable.

It is not necessary to maintain a constant word length throughout the processor. The word length and the quantization interval at each stage must be chosen so as to accomodate the expected dynamic range without excessive deterioration due to finite word length effects. The word length at each location in the processor must be minimized from the standpoints of economy and speed of operation. It is therefore of fundamental importance to design the processor in such a way that the dynamic range requirements are minimized.

SECTION 5 - DIGITAL PROCESSOR CONFIGURATIONS

5.1 INTRODUCTION

Several digital processor configurations are proposed in this chapter. The processors are based on two different concepts: decoding and pulse compression. The analog processor described in the previous chapter uses the decoding approach. The pulse compression method is based on converting the CW signal into a pulsed signal followed by standard pulse radar techniques. Also included are two hybrid processors where some analog processing is performed prior to A/D conversion. While reading the block diagrams, it must be remembered that all the video signals are made up of two separate signals, that is, the in-phase and the quadrature components of the original high frequency narrowband signal.

5.2 *PROCESSORS USING THE DECODING APPROACH*

5.2.1 Configuration D-1

A schematic of this processor is shown in Figure 5.1. It can be easily seen that this is a video frequency equivalent of the analog processor discussed in Section 3, where the clutter rejection filter has been replaced by a low-pass filter with a notch. This filter serves to increase the signal-to-noise ratio since it passes the desired signal while reducing the noise bandwidth from the original 5 MHz to 50 KHz. Because of the reduced bandwidth, the output of the filter may be resampled at a much lower rate, i.e., 50 KHz, without loss of information. Also, since clutter is attenuated by the notch filter, the word length requirement at the integrator is much lighter than that at the A/D converter.

If a recursive filter is used for the notch/LP filter, it suffers from the same disadvantage as the analog filter in that the settling time is very large. This will necessitate a weighting scheme before the filter and discarding the output

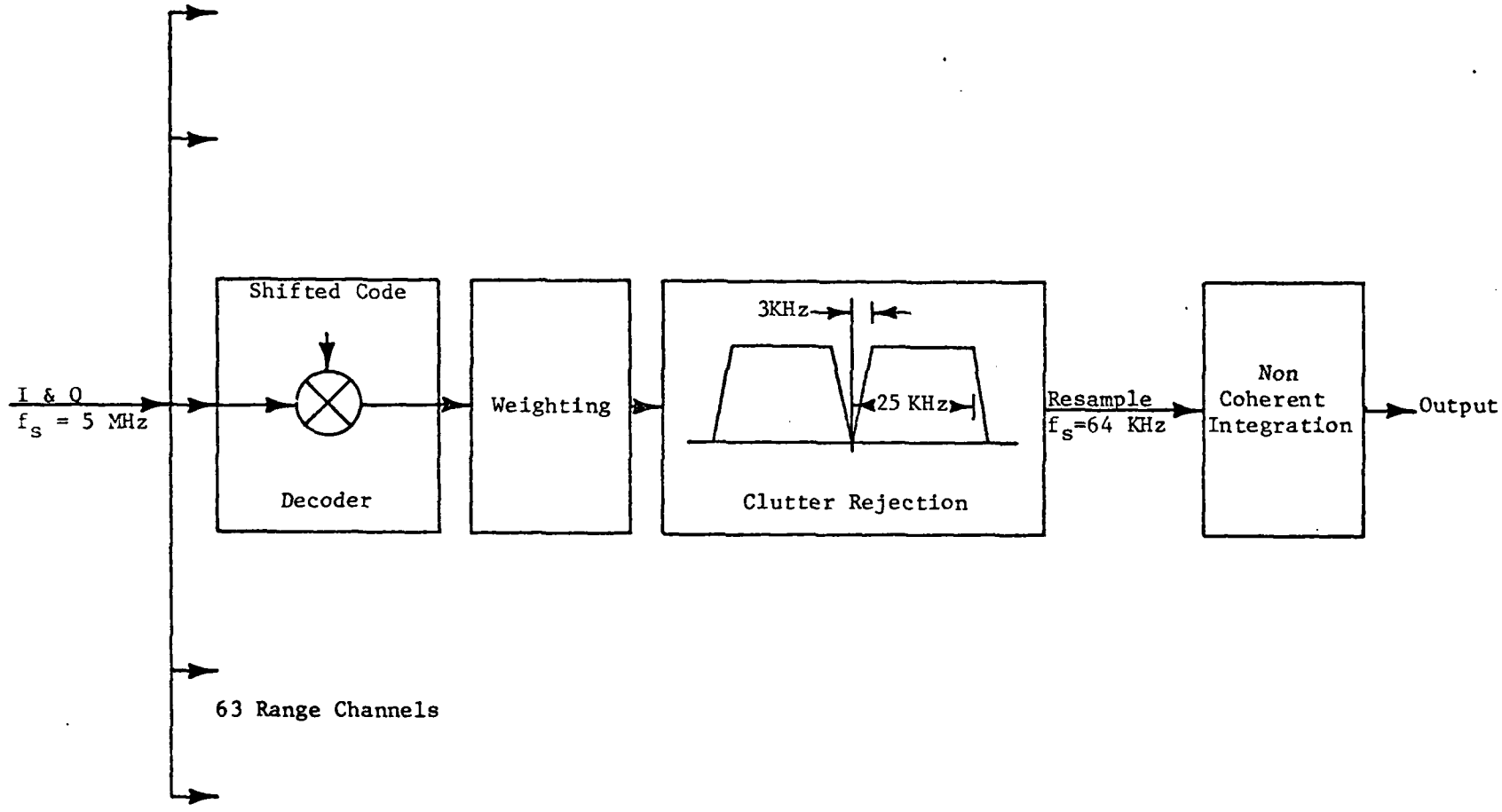


Figure 5.1 Configuration D-1

until the transient effects are negligible. If a non-recursive or a transversal filter is used, the transient response problem is almost completely eliminated. For example, if a L-tap filter is used, it is only necessary to discard the first L samples of the output to avoid transients. This is usually much fewer than the number of samples that must be abandoned when a recursive filter is used. In addition, if the number of taps is the same as the ratio of the input sampling rate to the reduced sampling rate contemplated, filtering and sampling rate reduction can be achieved simultaneously using a fixed window realization of the transveral filter. Such a filter can be implemented with just one multiplier and an accumulator. No signal weighting is necessary when non-recursive filters are used. Note that when weighting is required, it can be performed prior to decoding. Thus, the input can be weighted only once instead of once in each channel.

5.2.2 Configuration D-2

As can be seen from Figure 5.2, this is the same as D-1 except that non-coherent integration is replaced by coherent integration. This is achieved by using an FFT processor with 128 input samples, and a doppler resolution of 500 Hz corresponding to 2 msec look time. The obvious advantage of coherent processing is the additional improvement in signal-to-noise ratio as compared to the non-coherent case. Another advantage is the doppler resolution made available by the FFT processor. Even if doppler measurement is not contemplated, doppler separation may be useful in resolving range ambiguities. It is shown in Appendix A that the frequency response of the FFT processors feature large side lobes. Thus, if there are two targets in the same range cell but with different doppler shifts, the presence of the side lobes produces a large interference between the two target returns. It is therefore necessary to multiply the signal by a weighting sequence before fast Fourier transformation.

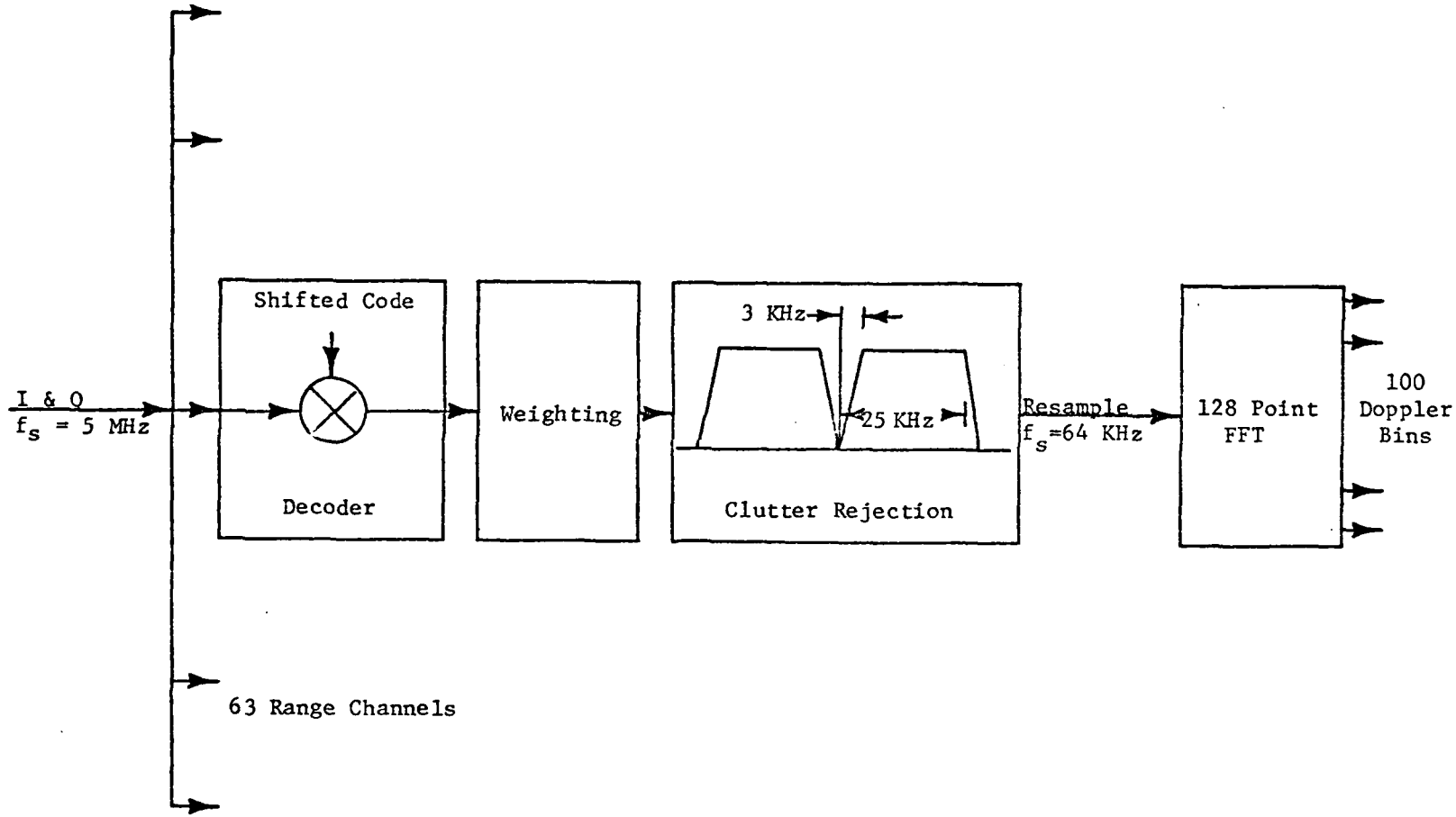


Figure 5.2 Configuration D-2

5.2.3 Configuration D-3

Since the FFT processor provides frequency resolution, the notch filter has been removed in this configuration shown in Figure 5.3. Separation of the target from clutter is entirely based on the difference in their doppler shifts. Eventhough transient response is no longer a problem, this processor suffers from several drawbacks. Since there is no clutter attenuation, the word length in the FFT processor must be large enough to accomodate the large ground clutter. The absence of the low pass filter dictates that the sampling rate at the FFT processor be maintained at 5 MHz. The doppler bins are still separated by 500 Hz and hence only 100 bins are required to cover the doppler range of interest. This means that the input to the FFT contains 10000 samples (2 m sec at 5 MHz sampling rate) while only the first 100 output samples are required. Thus, alternate and more efficient implementation of the FFT must be considered. The weighting requirement before FFT is even more critical than before because of the large amount of clutter entering the processor. The high sampling rate and the large word length requirement makes this processor very unattractive.

5.2.4 Configuration D-4

In an effort to reduce the word length requirement at the FFT processor, a delay line canceller is introduced in this configuration shown in Figure 5.4. Since the clutter spectrum has components at multiples of code repetition frequency, the delay in the canceller must be an integer multiple of the code period. It is shown in Figure 5.5 that a delay of two code periods is desirable since it has better gain characteristics for the doppler frequencies of interest. It is shown in Appendix B that the transient response of a two pulse canceller can be fully suppressed by discarding the first and the last output pulses. This corresponds to

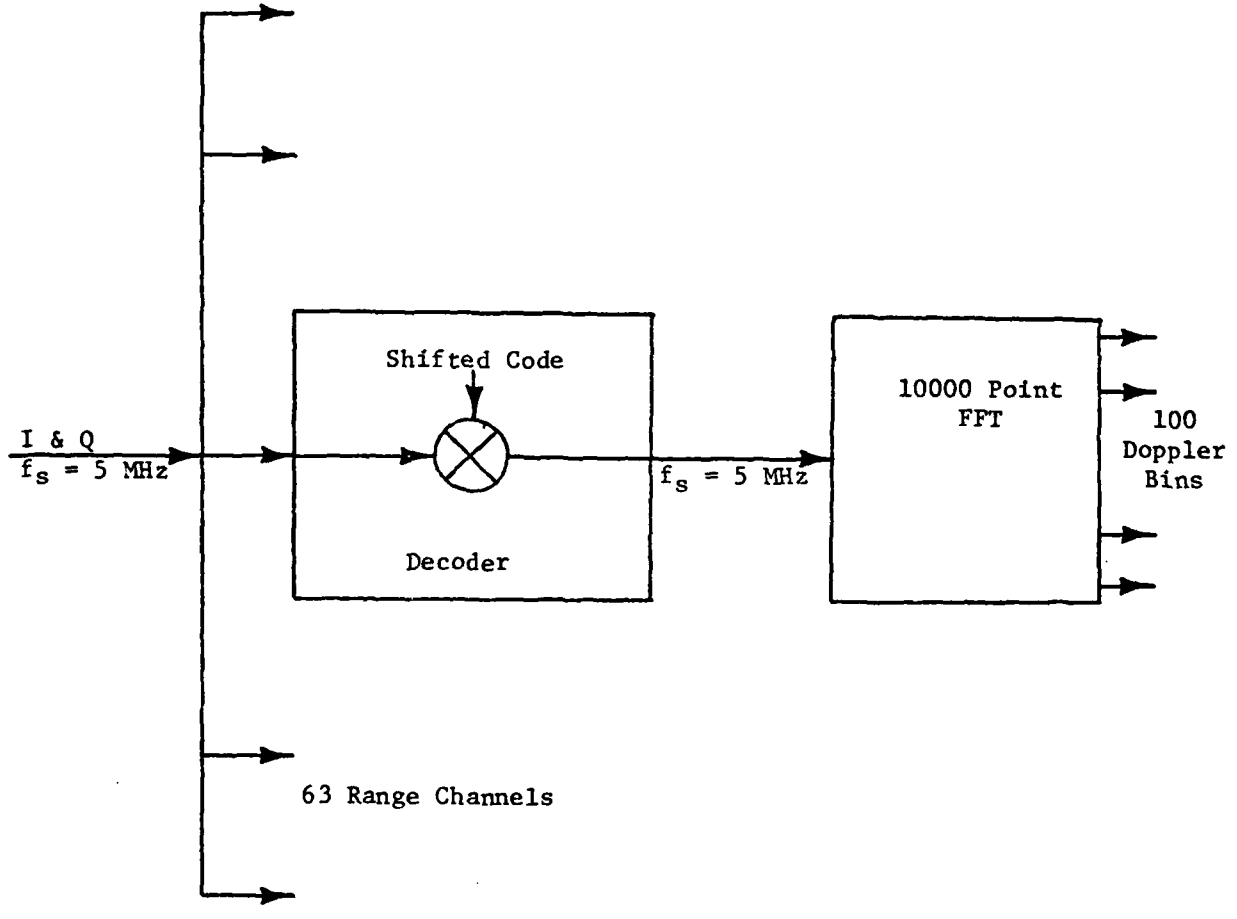


Figure 5.3 Configuration D-3

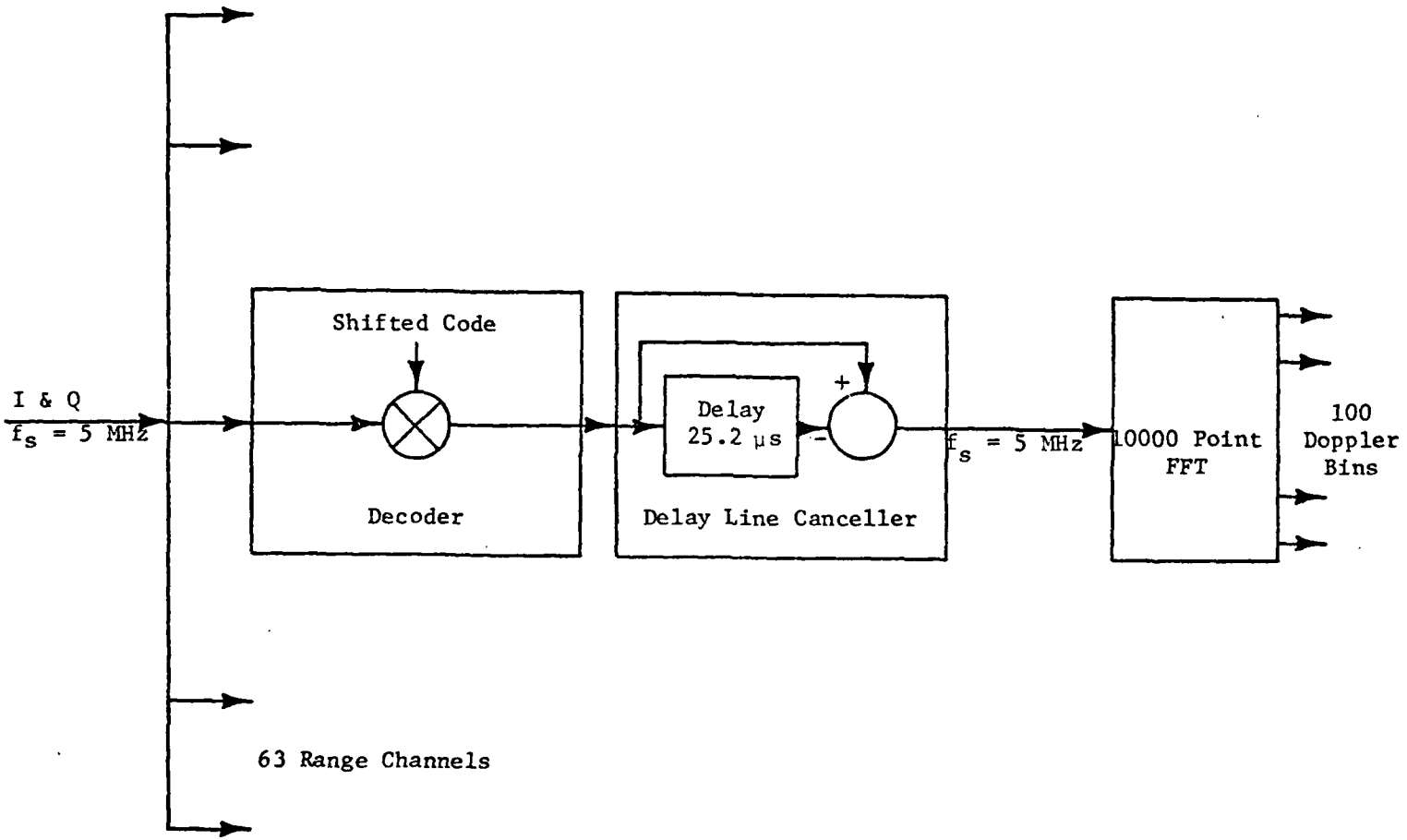


Figure 5.4 Configuration D-4

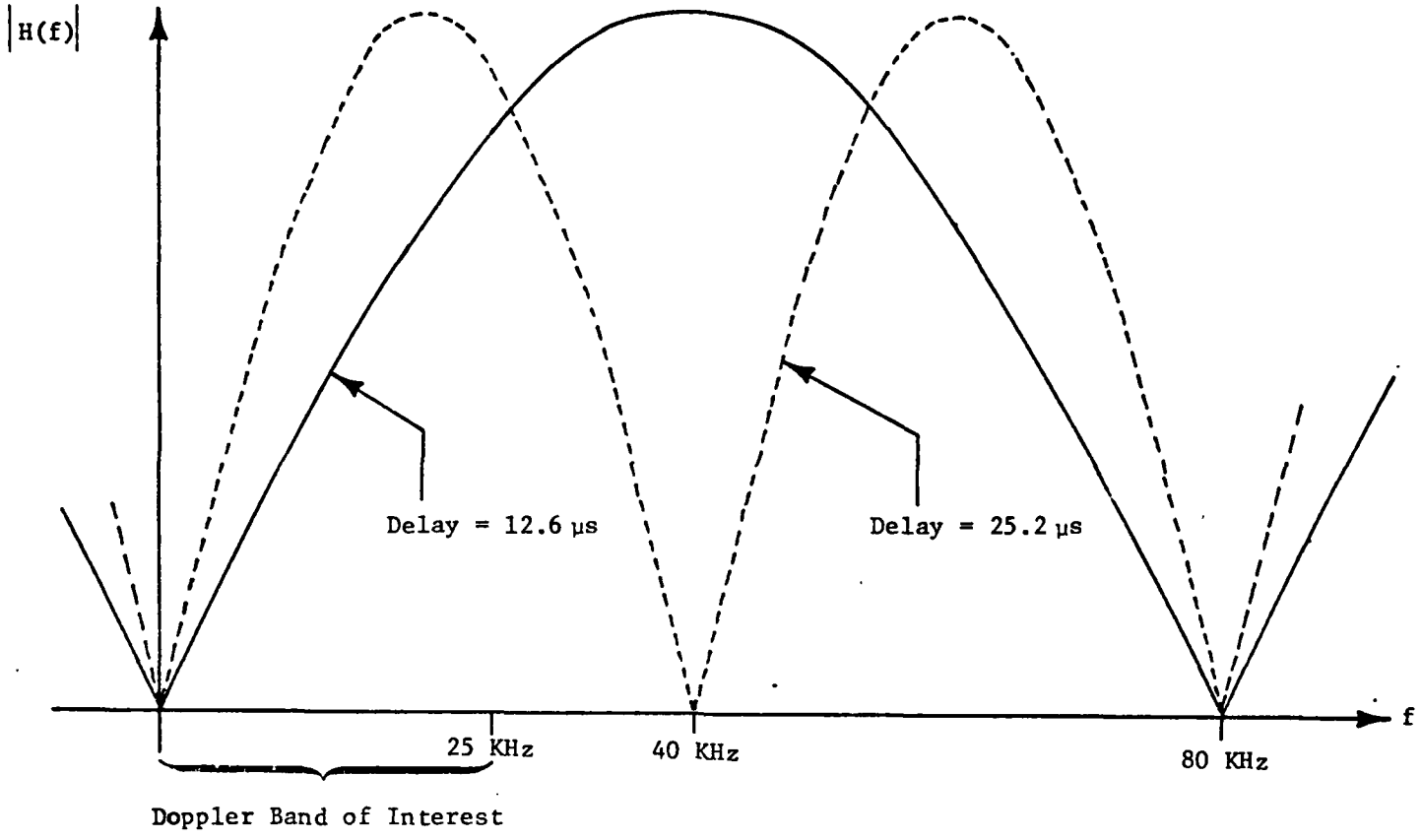


Figure 5.5 Magnitude Responses of Two Delay Line Cancellers

two code periods in the beginning and at the end of the output, which is a small portion of the total look time. Furthermore, it is shown in Appendix C that eventhough the MTI filter gain is not the same for all dopplers, the signal-to-noise ratios in the various FFT doppler bins are nearly the same. However, the signal level varies from one bin to the other and a different threshold must be chosen for each bin. The major disadvantage of this configuration is the fact that the FFT processor must operate with an input sampling rate of 5 MHz.

5.2.5 Configuration D-5

Due to the presence of enormous amounts of clutter, large word lengths are required in the processor until the clutter rejection filter. It is therefore advantageous to perform clutter suppression as early in the processor as possible. In this configuration shown in Figure 5.6, the delay line canceller is implemented immediately after A/D conversion. It is shown in Appendix D that the two pulse canceller before or after decoding are equivalent provided the delay is an integer multiple of the code period. In this case, the performance of D-4 and D-5 are identical. However, only one canceller is needed in D-5 as opposed to one in each range channel in D-4. Also, the early reduction in dynamic range could lead to a more economical implementation.

5.2.6 Configuration D-6

Shown in Figure 5.7, this configuration is the same as D-3 except that a low pass filter is introduced prior to FFT processing in order to reduce the sampling rate. Eventhough the number of samples into the FFT processor is greatly reduced, there is no loss in the processor gain since low pass filtering provides bandwidth reduction and hence an increase in the signal-to-noise ratio. Due to the absence of a clutter rejection filter, large amounts of clutter enter the FFT processor thus requiring a large word length. Furthermore, due to the sidelobes inherent

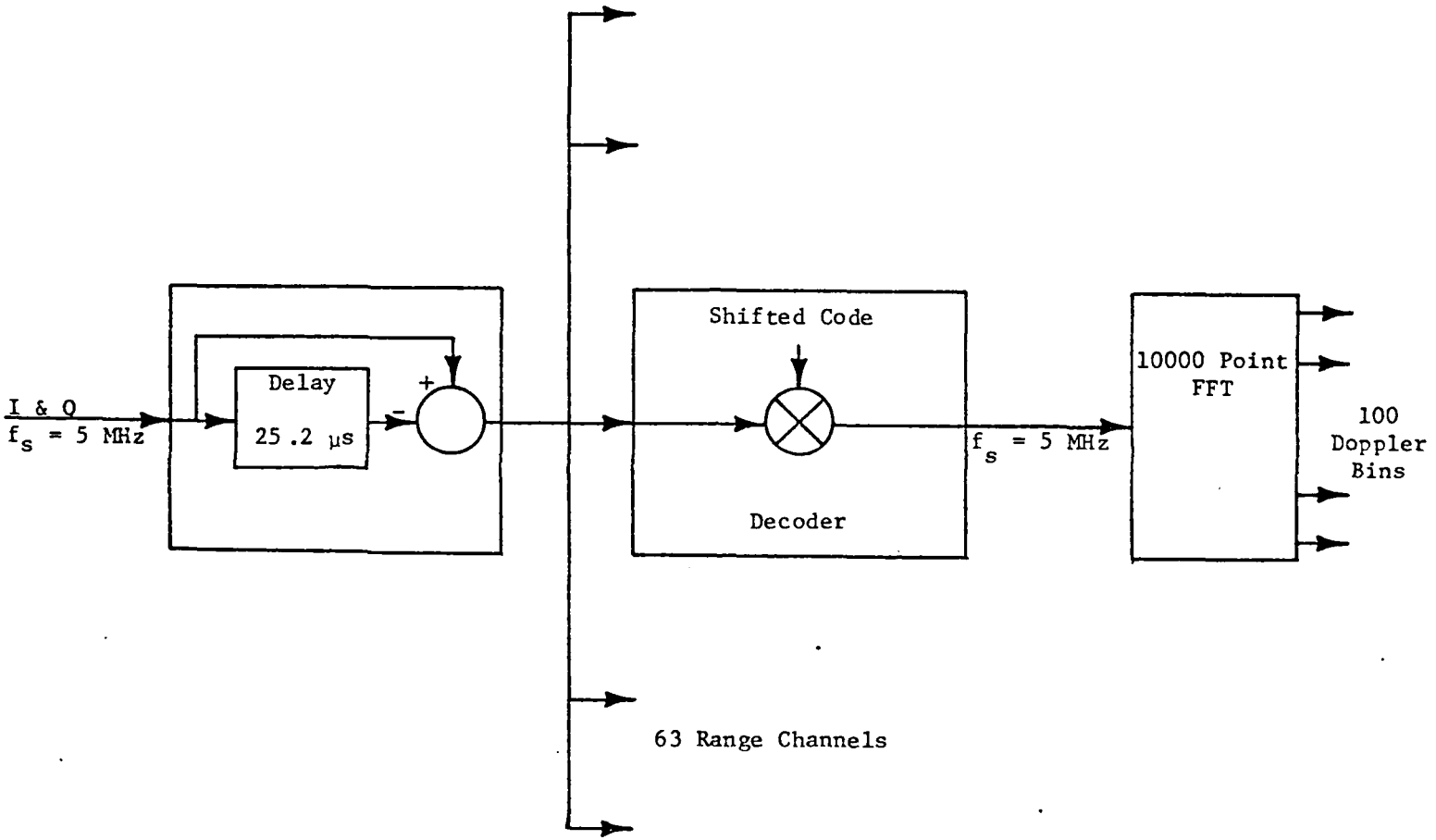


Figure 5.6 Configuration D-5

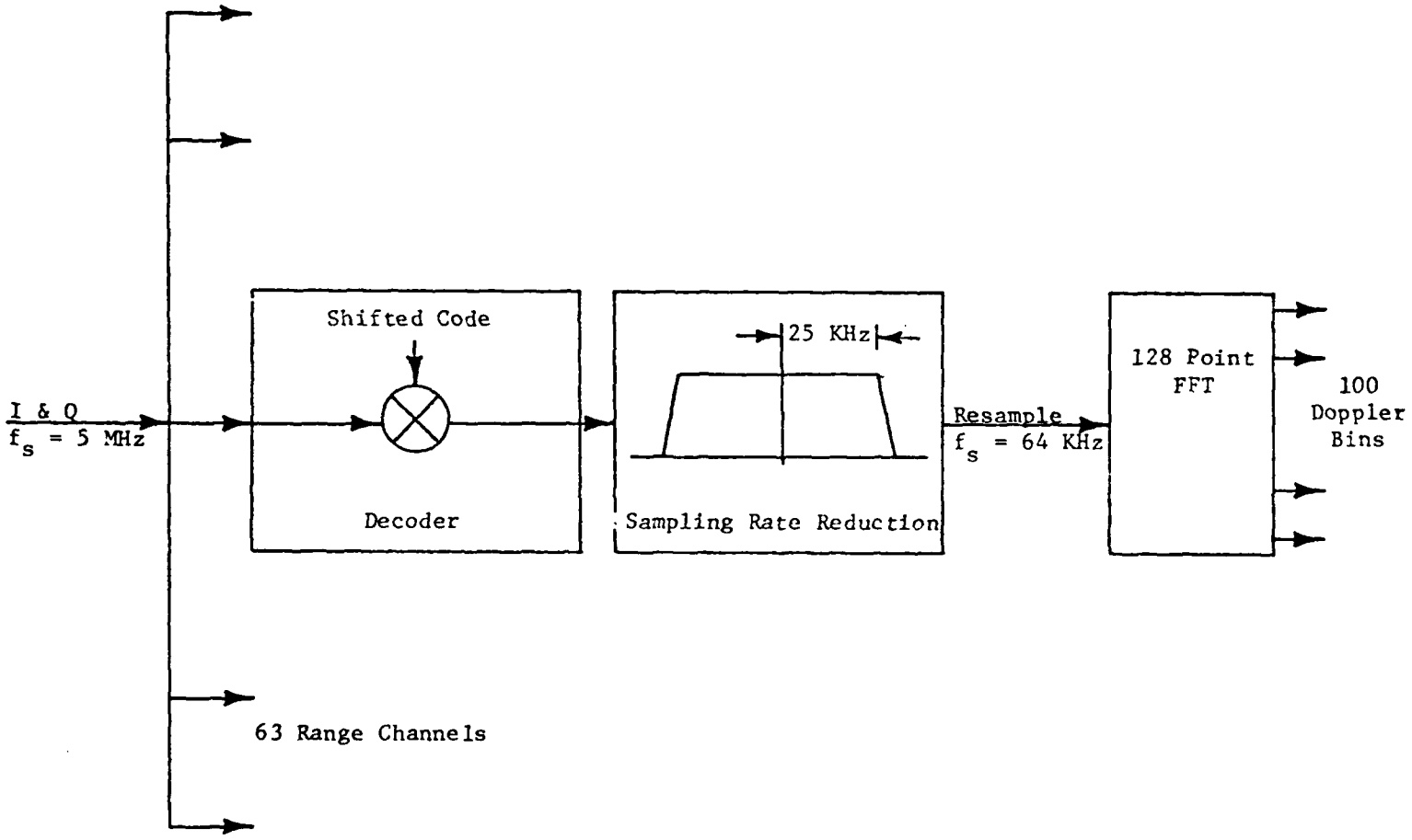


Figure 5.7 Configuration D-6

in FFT processing, the large clutter component will leak into other doppler bins unless the input is properly weighted.

5.2.7 Configuration D-7

This processor, shown in Figure 5-8, combines the advantages of configurations D-4 and D-6 in that both a clutter rejection filter and a sampling rate reduction filter are utilized. Here again, the low pass filter may be recursive or non-recursive, and, if possible, filtering and sampling rate reduction may be combined by implementing a fixed window transversal filter. Such a filter contains only one multiplier and one accumulator in each range channel as opposed to several multipliers for any other type of filter realization. As in the case of Configuration D-2, weighting the input signal is required to reduce interference between two target returns in the same range cell.

5.2.8 Configuration D-8

This is exactly the same as the previous processor except that, as shown in Figure 5.9, the clutter rejection filter is implemented before decoding. Only one delay line canceller is needed as opposed to 63 in Configuration D-7.

5.2.9 Configuration D-9

The main motive behind this configuration is the possibility of using a very small word length (as small as one bit) at the decoder. In order to be able to achieve this, the doppler information must be extracted before the decoding operation. In this configuration shown in Figure 5.10, the incoming signal is first divided into several doppler channels and range resolution is later achieved in each doppler channel. The block named compensator is a complex weighting of the signal, and it is different for each doppler channel as described in Appendix E. Also shown in

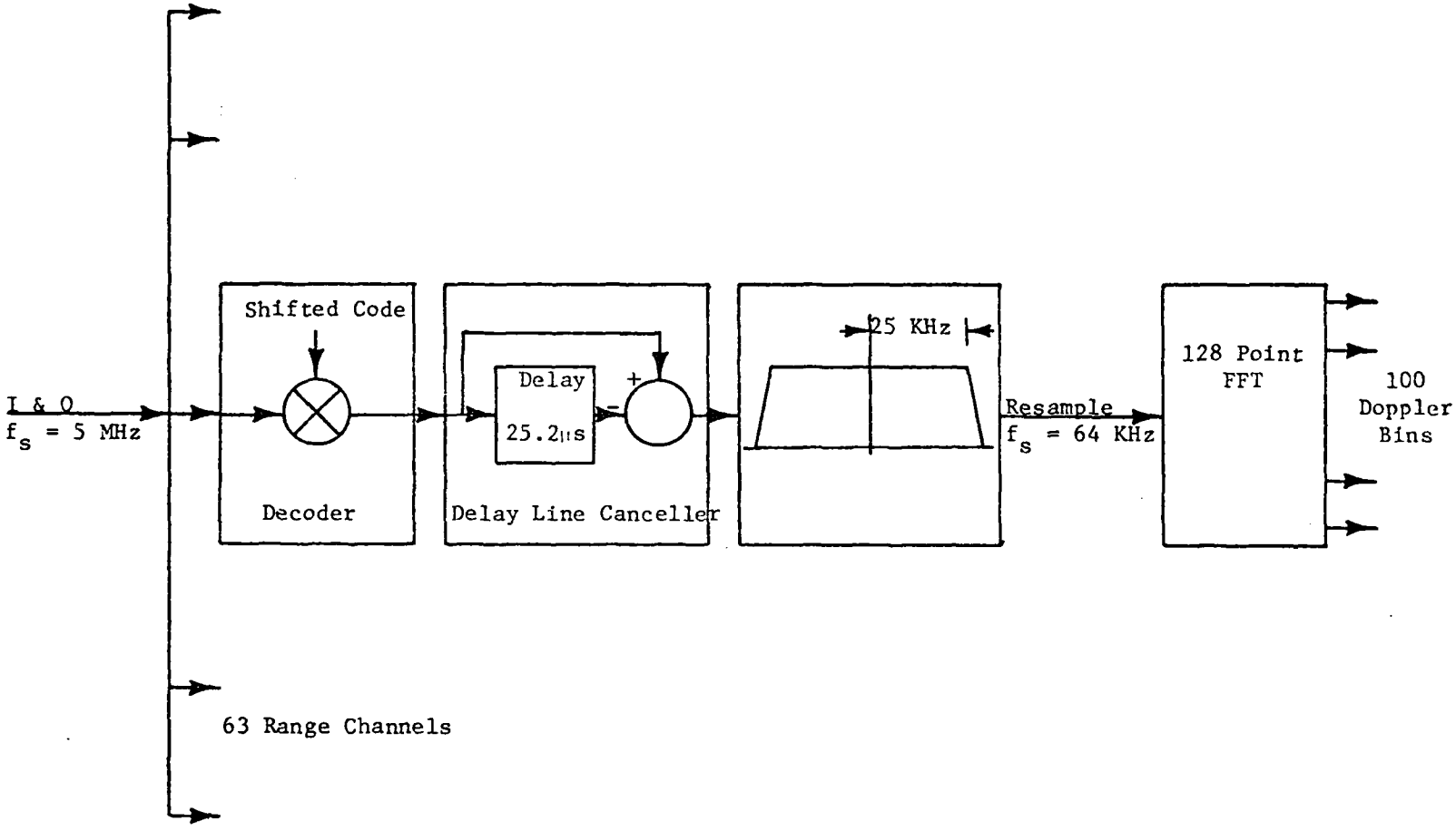


Figure 5.8 Configuration D-7

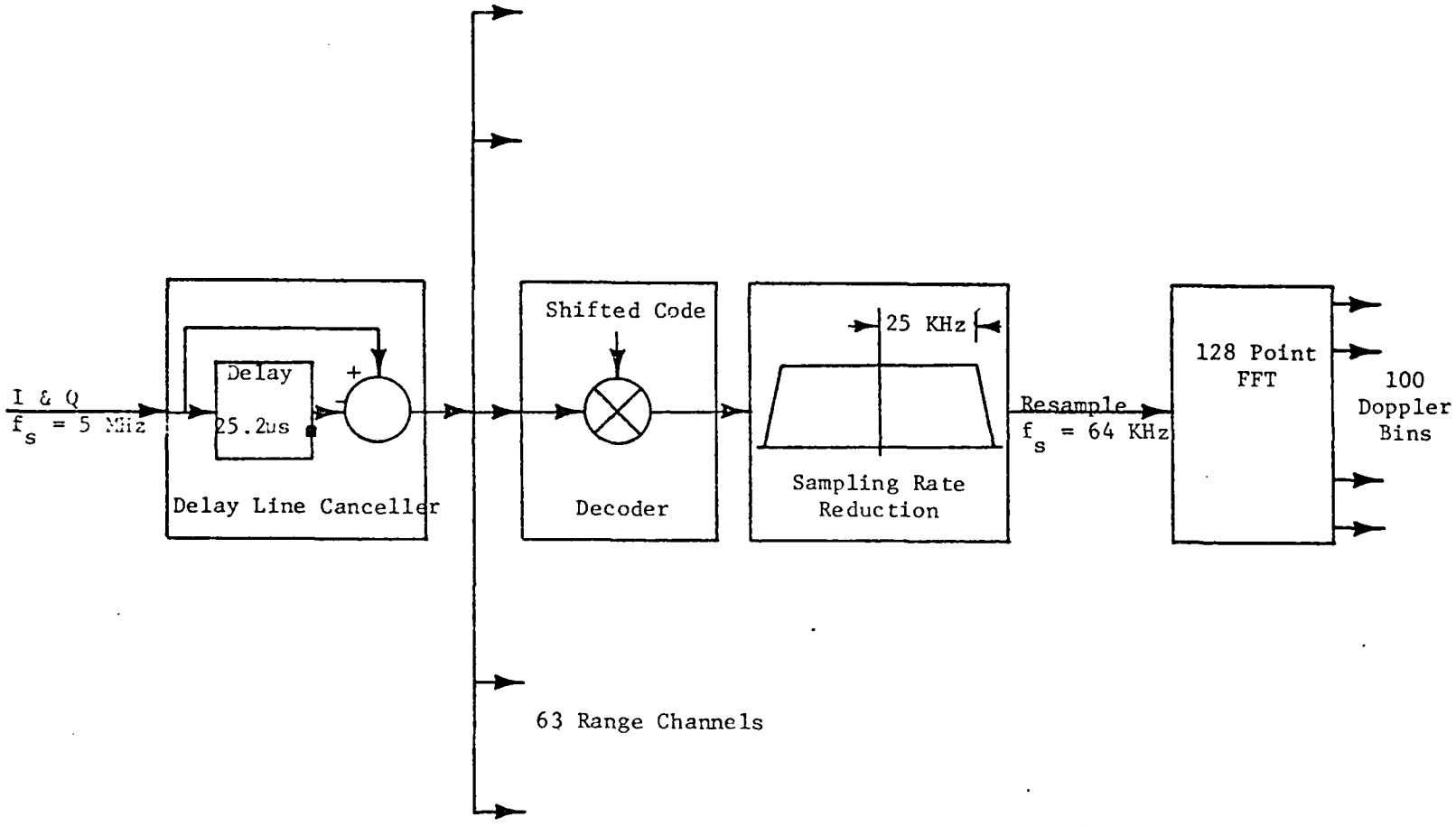


Figure 5.9 Configuration D-8

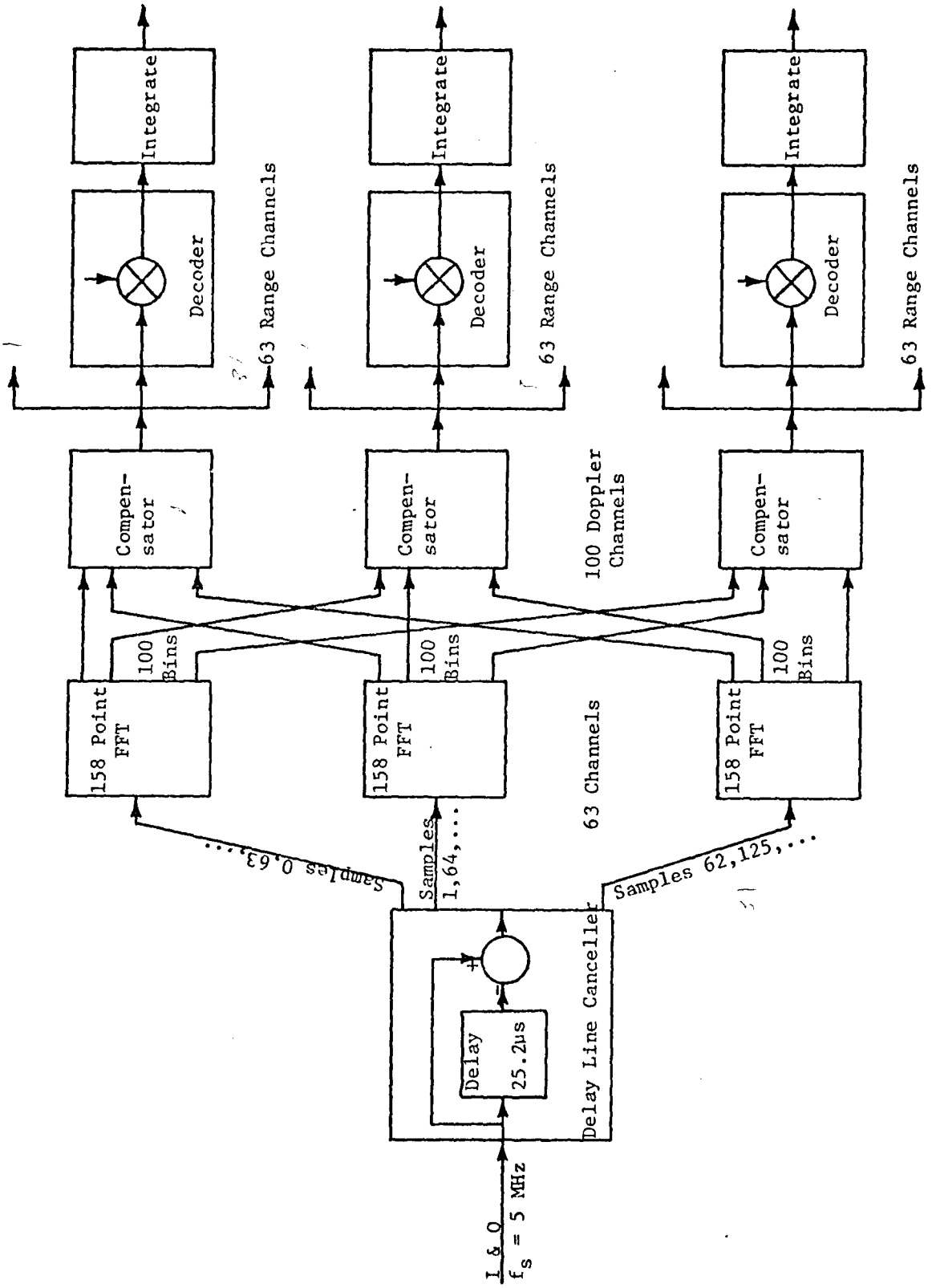
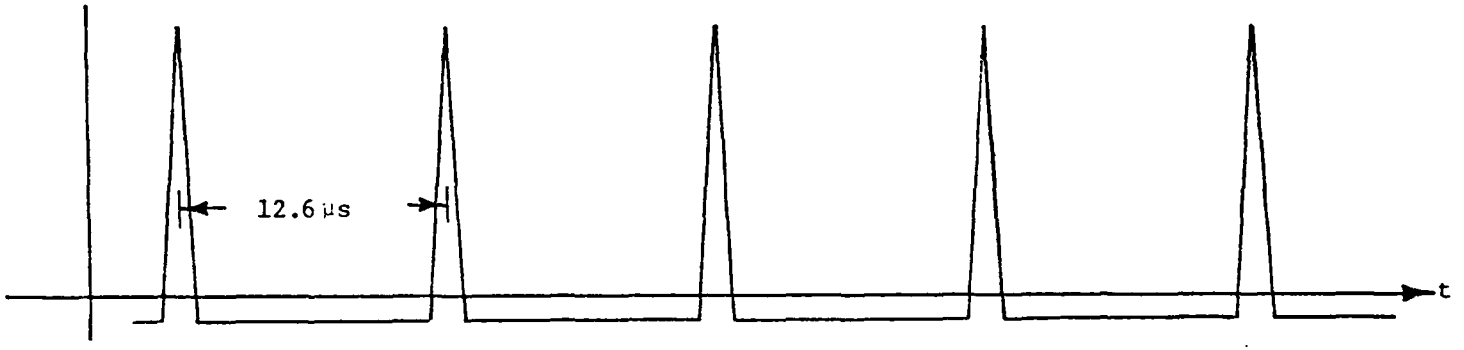


Figure 5.10 Configuration D-9

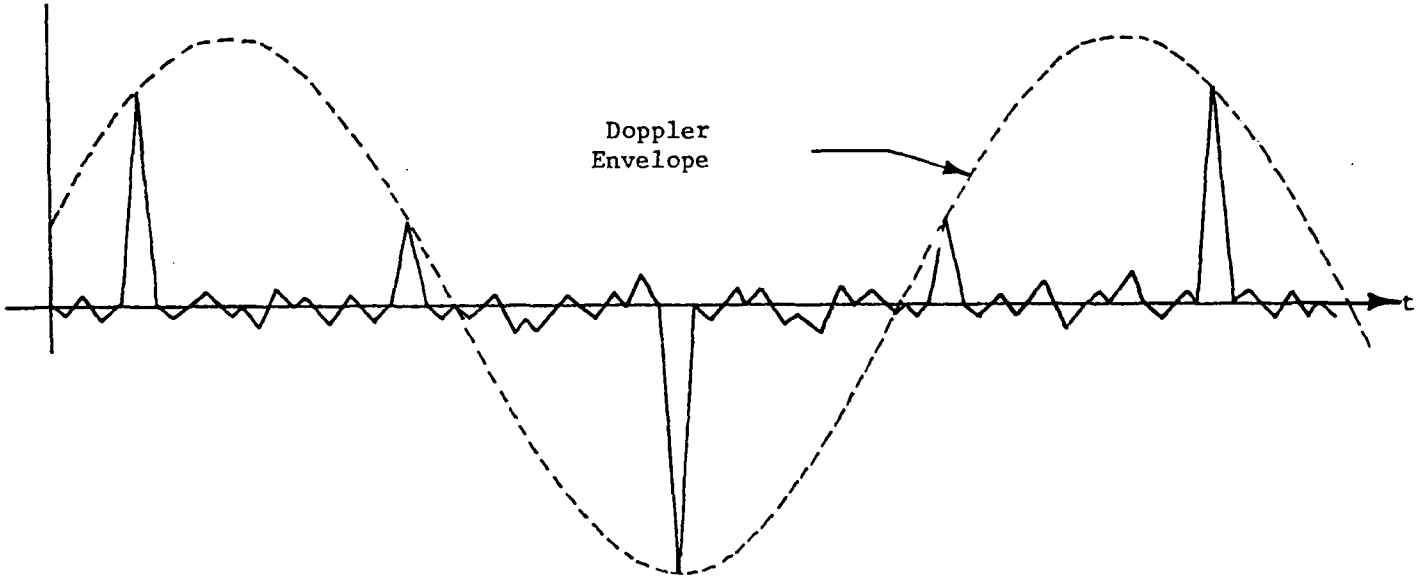
Appendix E is the fact that this configuration is identical to D-5 in its performance. The gain in signal-to-noise ratio is partially achieved by the FFT processor and the rest of the gain is attained during integration following the decoder.

5.3 PROCESSORS USING PULSE COMPRESSION

The pulse compression approach is an attempt to convert the CW signal into a pulsed signal, so that standard pulse radar processing techniques may be used. The pulse compression filter is a matched filter, which is matched to one period of the code. In the absence of any doppler, the output of the matched filter due to the target return is the same as the autocorrelation of the code. In the presence of doppler shift, the doppler envelope is impressed on the output as shown in Figure 5.11. However, because of the doppler mismatch (matched filter is matched to zero doppler) the output is no longer the autocorrelation but features larger range sidelobes, thereby reducing range resolution. The mismatch also reduces the main peak resulting in a weaker detection performance. Two configurations PC-1 and PC-2 are shown in Figures 5.12 and 5.13, the only difference between the two being the location of the clutter rejection filter. It is shown in Appendix F that the output of the pulse compression processor is not the same as that using the decoder except at zero doppler. It must be noted that the implementation of the processors using pulse compression is considerably simpler than those using decoders because fewer components are required. It must be noted too that a waveform ideally suited for the decoder approach may not be suitable for the pulse compression approach and vice versa.



(a) Output With No Doppler Shift



(b) Output Due to Doppler Shifted Signal

Figure 5.11 Matched Filter Output With and Without Doppler Shifts

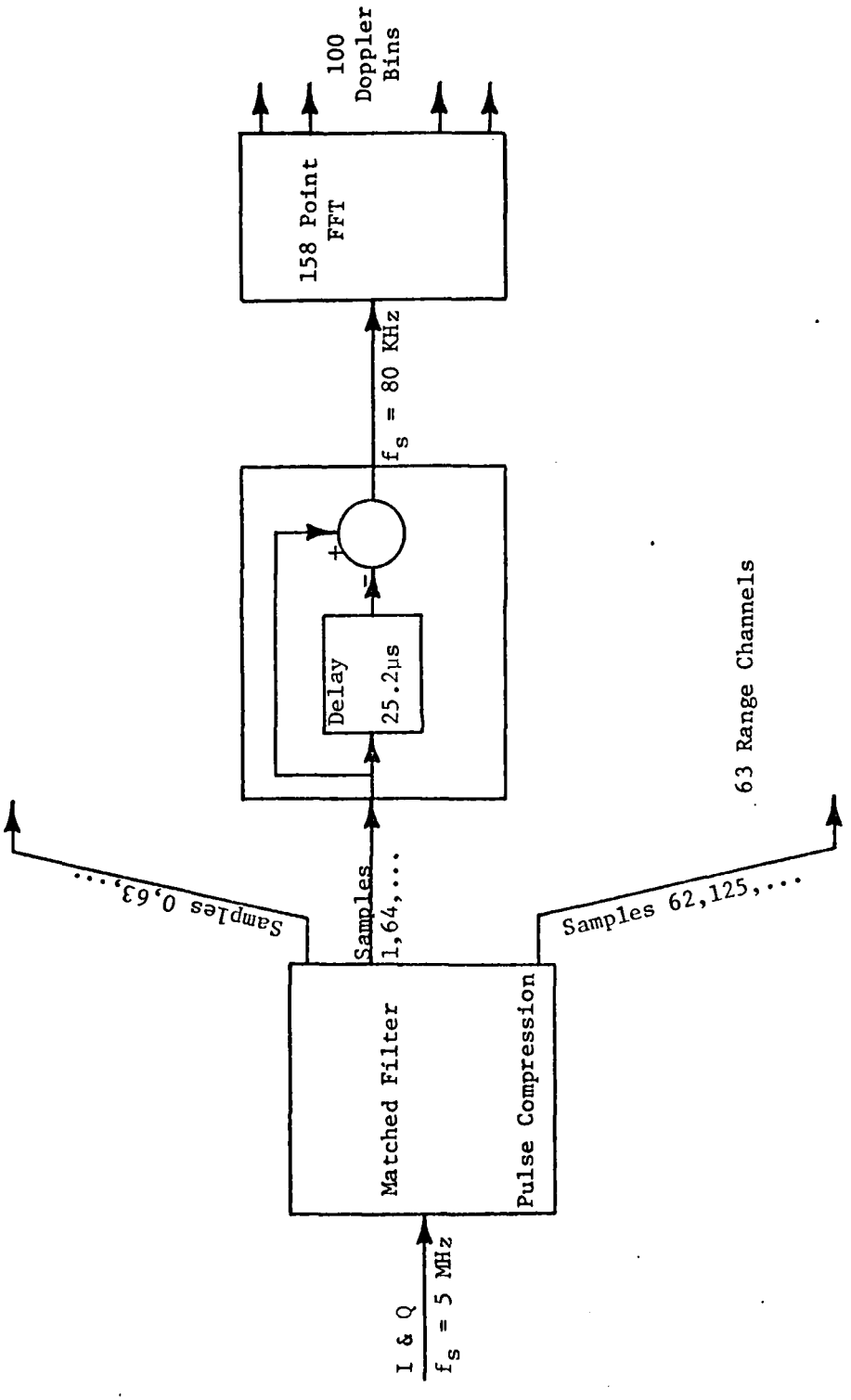


Figure 5.12 Configuration PC-1

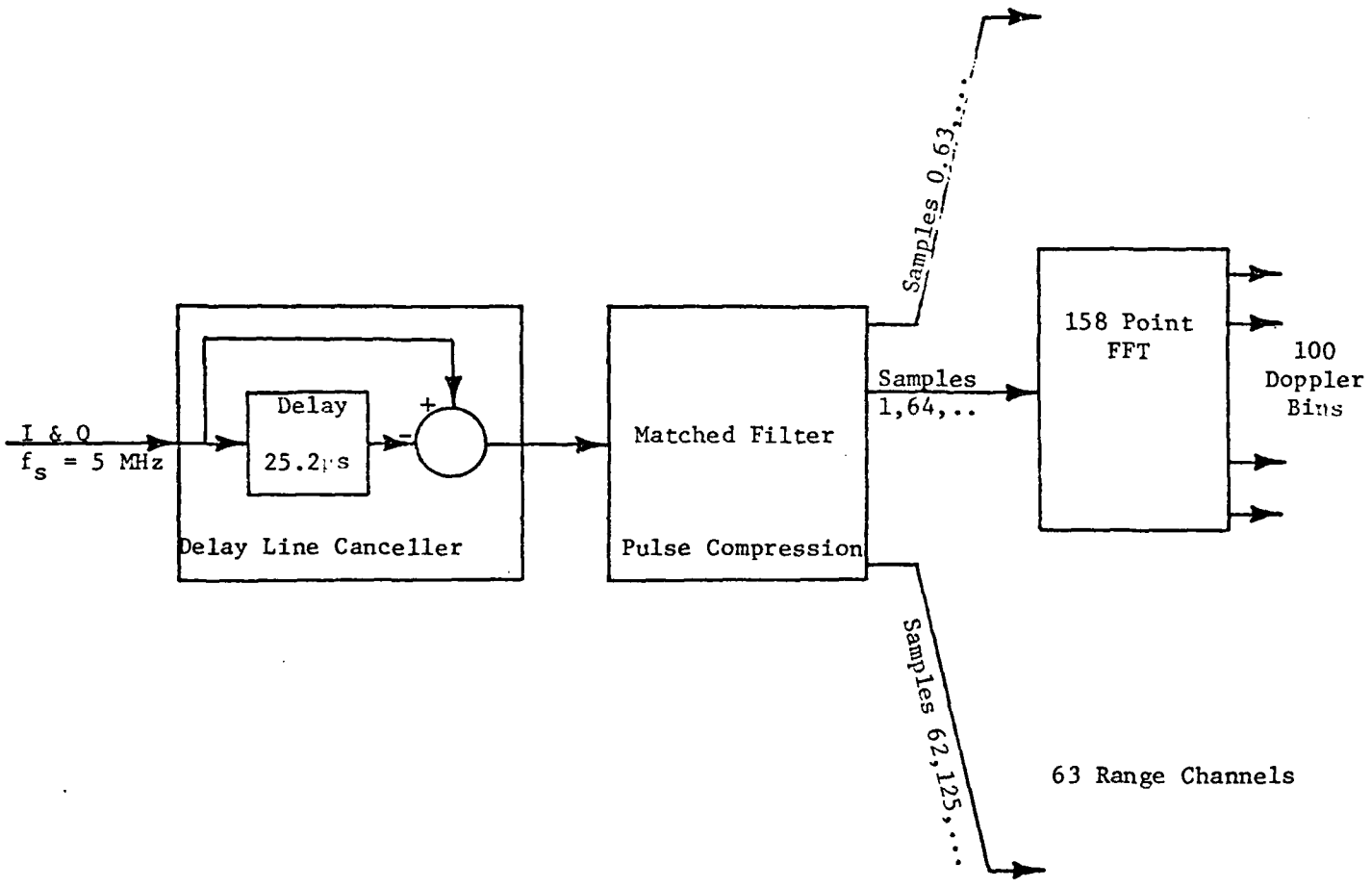


Figure 5.13 Configuration PC-2

5.4 HYBRID PROCESSORS

The principal limitations of the digital processors are the conflicting requirements of high sampling rate and large word length, while their advantages are flexibility and easy realization of coherent integration using FFT processors. The hybrid processors are designed to maintain these advantages while overcoming some of the drawbacks of digital processing by performing a portion of the processing before A/D conversion. Coherent integration capability of digital processing is utilized in Configuration H-1 shown in Figure 5.14. Because of clutter attenuation and bandwidth limiting by the analog filter, both the requirements of sampling rate and word length are made less stringent. The analog filtering may be performed either at video on the in-phase and quadrature signals or at IF. If IF filtering is used, the in-phase and quadrature components must be extracted prior to A/D conversion. The main disadvantage of this processor is the same as that of the analog processor described earlier, that is, the loss in achievable processor gain due to the large settling time of the notch filter. This problem can be alleviated by using Configuration H-2, shown in Figure 5.15, where clutter rejection is performed in the digital portion. This is accomplished by the use of a delay line canceller in which the transient response problems are completely avoided. However, since no clutter attenuation takes place before A/D conversion, large word lengths are required to accomodate the expected dynamic range. This problem is not as severe as in the all-digital processors because of the availability of A/D converters with large word lengths at lower sampling rates. It must be noted here that eventhough the sampling rate has been reduced, the number of A/D converters has been increased from one to one in each range channel. Superiority of hybrid processing is accompanied by a loss in flexibility due to the fact that the analog portion of the processor cannot be easily adapted to any changes in the waveform or other parameters.

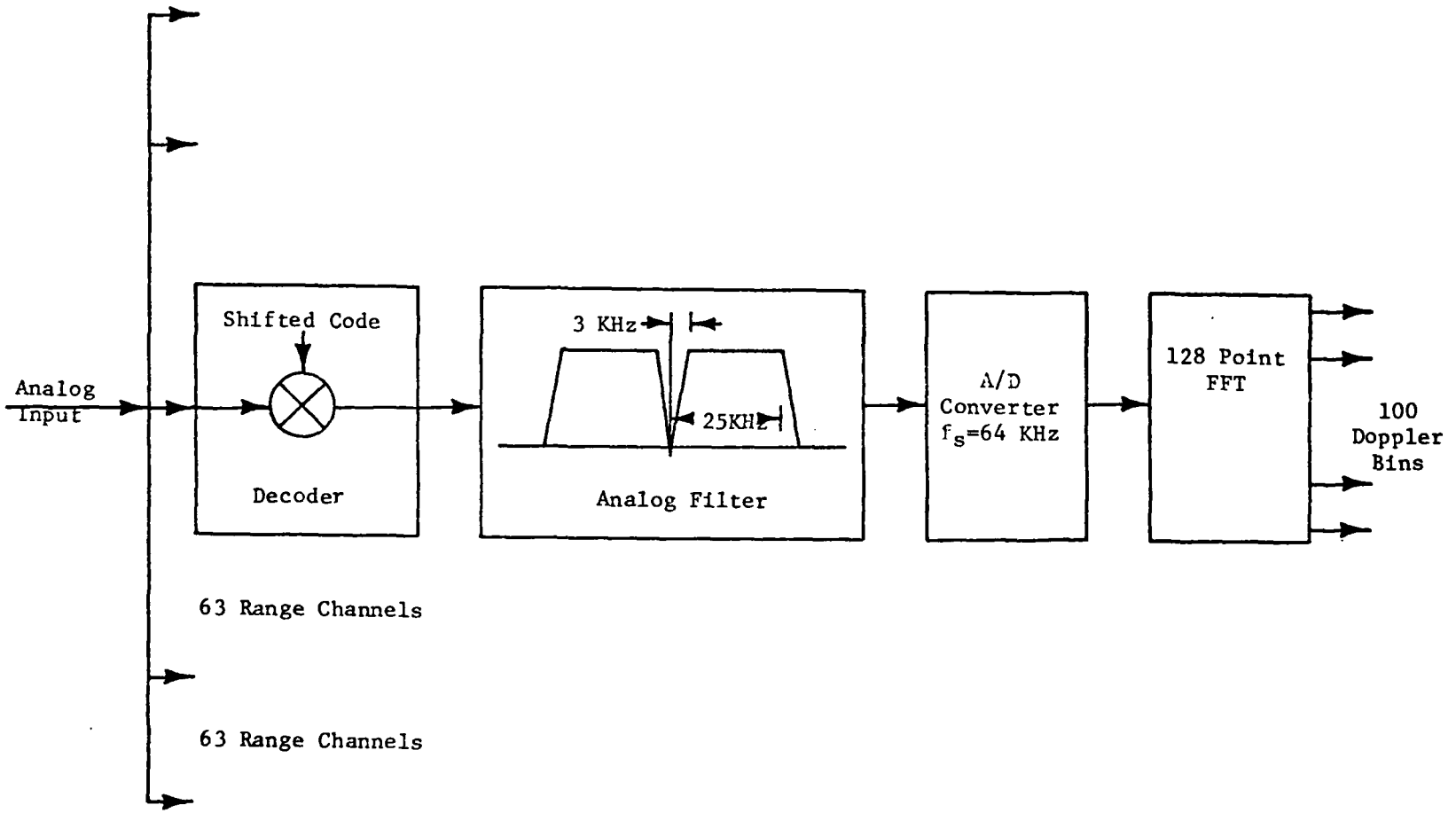


Figure 5.14 Configuration H-1

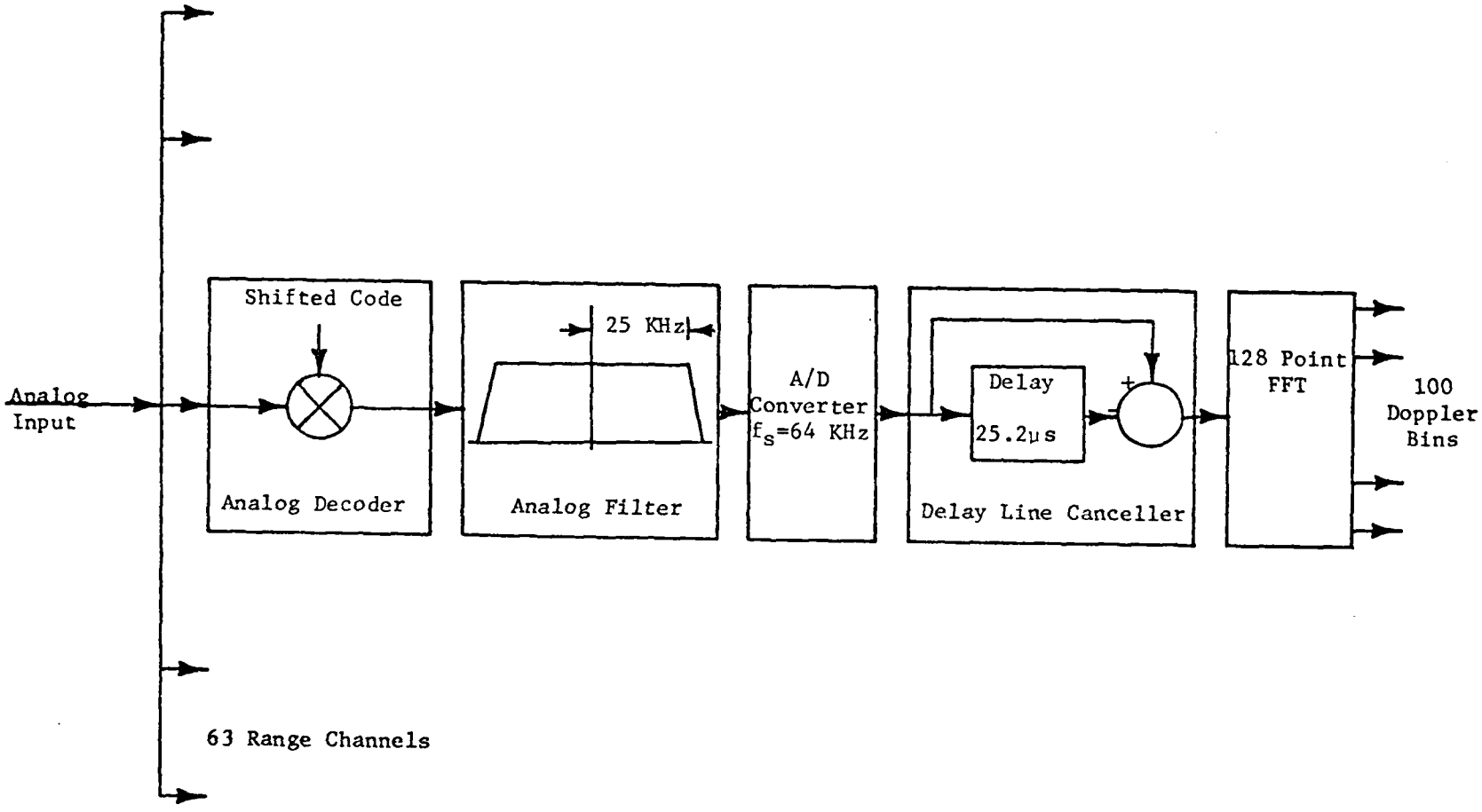


Figure 5.15 Configuration H-2

5.5 CONCLUDING REMARKS

The principal attractions of digital processors are the flexibility they offer and the possibility of obtaining filter characterics which could not be realized by analog systems. In this chapter, several configurations were proposed as possible candidates for digital processors in a phase coded CW radar. Each processor is based on one of two different concepts: decoding and pulse compression. Even-though several of the configurations are mathematically equivalent, their detection performance may vary because of the nonlinearity inherent in quantization. Therefore, each of th processors must be analyzed in detail to minimize the cost for specified performance levels.

Figure 5.16 shows the uncertainty function for a PN sequence of length 63, a clock rate of 5 MHz and a look time of 2 m sec. As can be seen the signal exhibits the desirable properties of good range and doppler resolution and small range side lobes. These properties can only be attained if the processor consists of a bank of filters each matched to a different doppler shift. Configuration D-3 is, in fact, such a processor while the other decoder configurations using coherent integration are close approximations. However, the processors using pulse compression produce large range side lobes because of the doppler mismatch in the pulse compression filter which is matched to zero doppler. This can be seen from Figure 5.17 which shows the magnitudes at the output of the pulse compression filter for various dopplers. The input to the pulse compression filter is a doppler shifted phase code at zero range delay. At zero doppler, the range side lobes are exactly the same as in the decoder approach, since the pulse compression filter is exactly matched to the input waveform. However, as the doppler shift of the input increases, the mismatch produced manifests itself in large range side lobes and a reduced main lobe. While this effect is not pronounced at small doppler shifts, the degradation is unacceptably large at higher doppler frequencies. It must also be noted that the

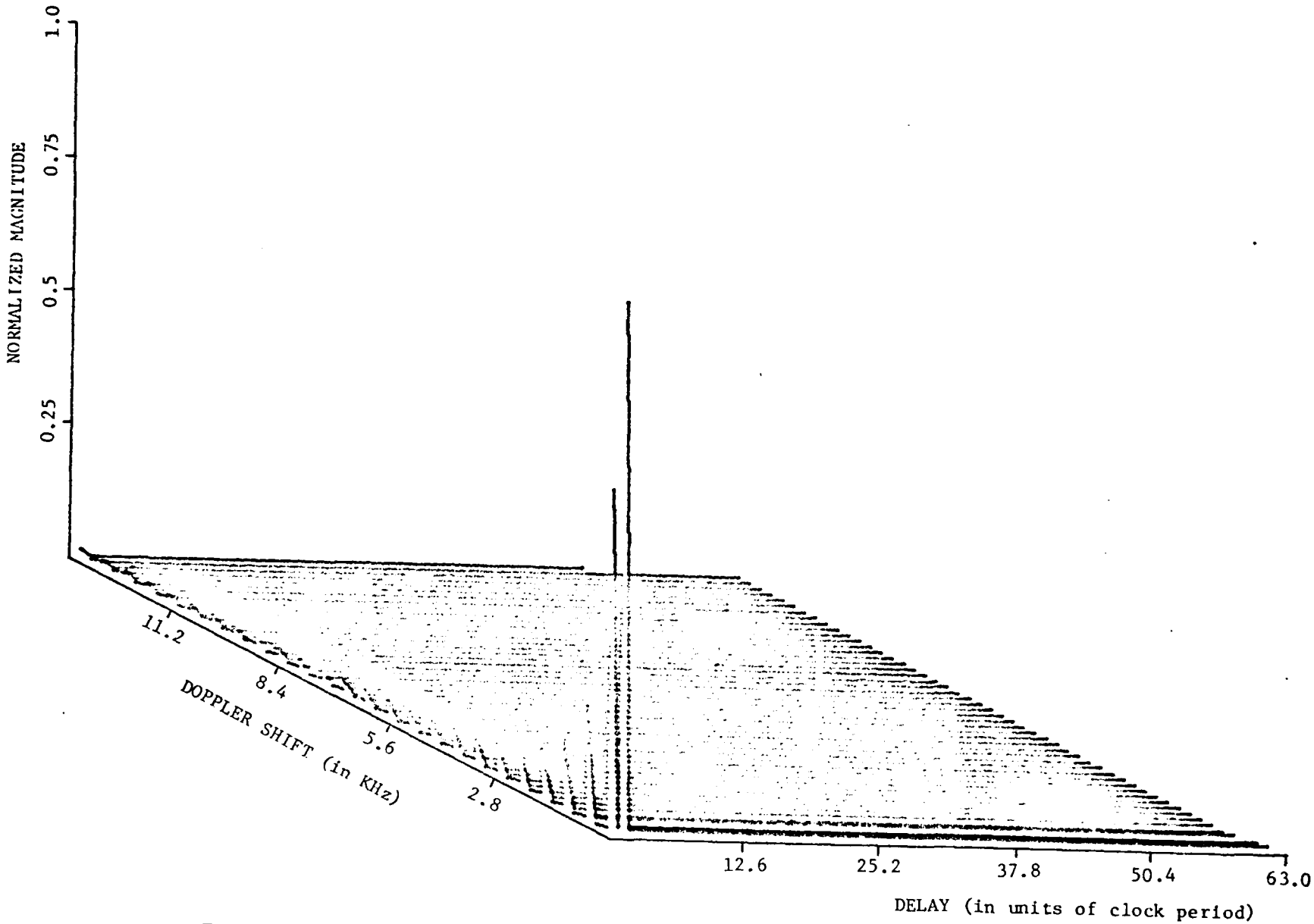


Figure 5.16 Uncertainty Function for a Periodic PN Sequence of Length 63

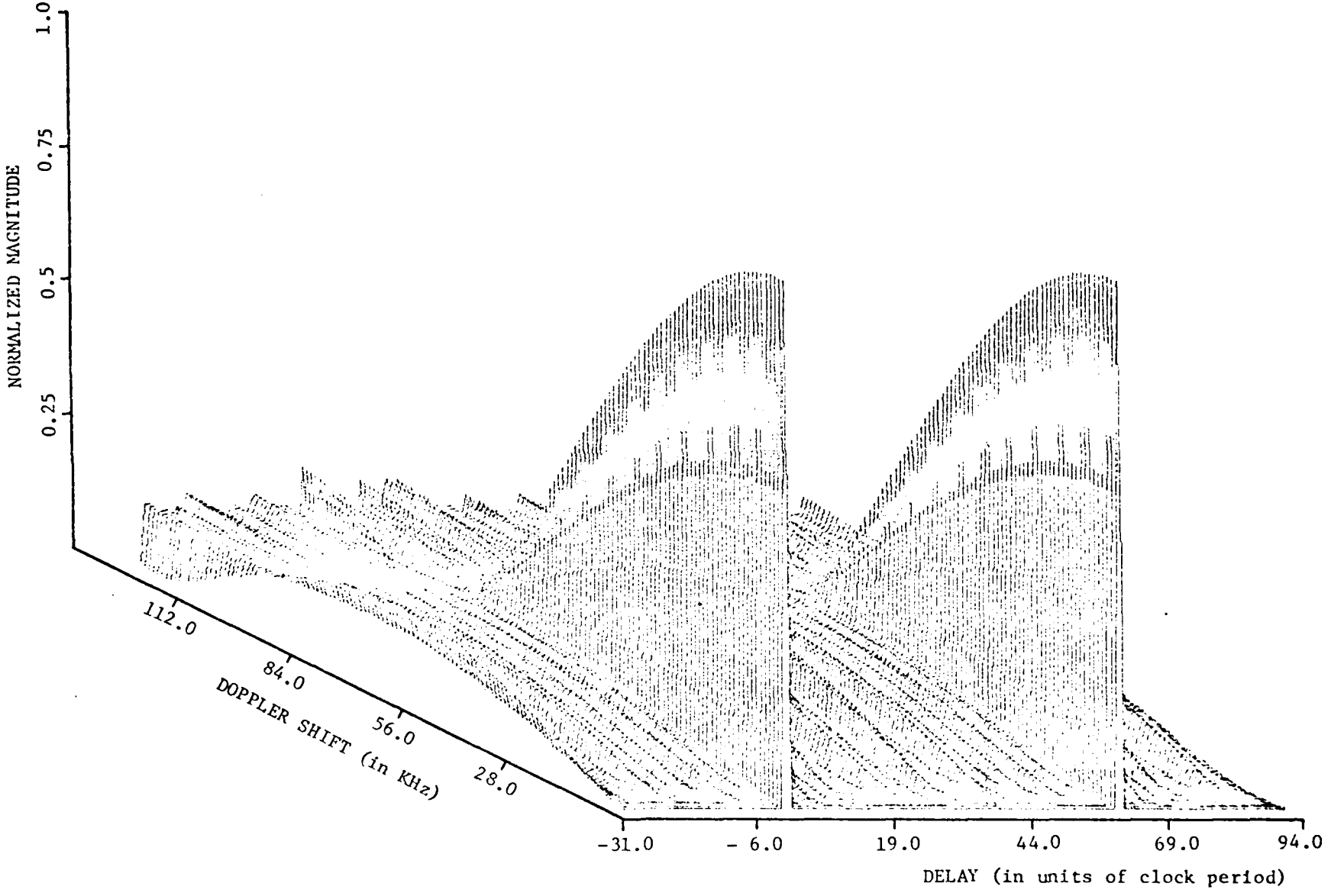


Figure 5.17 Magnitude at the Output of the Pulse Compression Filter

range side lobe structure depends upon the initial shift in the code (initial contents of the shift register) unlike in the decoder approach. The severity of the range side lobes is entirely dependent upon the modulating waveform and it is necessary to analyze several waveforms in order to select one which is suited for the pulse compression processor.

The main factor that restricts the performance of a digital processor is the effect of finite word lengths. This effect can be reduced by expanding the word length, but an upper limit on the word length is imposed by the high sampling rate necessary to avoid errors due to aliasing. In an effort to reduce the sampling rate required, two hybrid processing schemes were proposed where some flexbility is sacrificed by performing a portion of the processing before analog-to-digital conversion.

SECTION 6 - SIMULATION AND RESULTS

6.1 INTRODUCTION

In order to study the performances of the various configurations, digital computer simulations of the processors have been developed. To facilitate a Monte Carlo analysis of the performance, computer programs have also been developed to generate samples of synthetic video return including ground clutter, noise and return from a single moving target. All the programs are written in FORTRAN compatible with the CDC 6600 computer system. Program listings are included in Appendix H.

6.2 SYNTHETIC VIDEO RETURN

Samples of the video return are normalized so that the power in the target return is unity. It is assumed that the sampling rate is equal to the clock rate of the PN sequence. The total return includes contributions due to the target, noise and clutter.

6.2.1 Target Return

The target is assumed to be a moving target with a doppler frequency f_d and a range delay corresponding to k clock periods. Thus, if the target range is R and the range resolution is ΔR, then the range delay in clock periods is

$$k = \text{Int } (R/\Delta R) \text{ modulo } (N) \qquad (6.1)$$

where Int $(\cdot)$ denotes "integer part of" and N is the length of the code. If the set $\{p(i)\}$ denotes samples of the periodic code, then the i^{th} sample of the in-phase and quadrature components of the target return are given by

$$x_I(i) = p(i-k) \cos (2\pi f_d i\delta + \psi_t) \quad (6.2)$$

$$x_Q(i) = p(i-k) \sin (2\pi f_d i\delta + \psi_t) \quad (6.3)$$

where ψ_t is a random phase which is constant over a look period. As can easily be seen, the total power in the target return is unity.

6.2.2 Noise

If the required signal-to-noise ratio in dB is SNR, then the total noise power is

$$\sigma^2 = 10^{-0.1\ SNR} \quad (6.4)$$

This noise power is divided equally between the in-phase and quadrature channels. The noise samples in each channel are assumed to have a Gaussian distribution with zero mean and a variance of $\sigma^2/2$. The samples are independent of one another and the noise sequences in the two channels are assumed to be independent of each other.

6.2.3 Ground Clutter Return

If the required signal-to-clutter ratio in dB is SCR, then the clutter power is

$$A^2 = 10^{-0.1\ SCR} . \quad (6.5)$$

Ground clutter return is assumed to have a steady or DC component and a random fluctuating or AC component. The i^{th} sample of clutter return in the in-phase and quadrature channels are generated according to the following equations.

$$c_I(i) = A\ p(i-k) [g_o \cos \psi_c + g_a f_I(i)] \quad (6.6)$$

$$c_Q(i) = A\ p(i-k) [g_o \sin \psi_c + g_a f_Q(i)] \quad (6.7)$$

where $\{p(i)\}$ is the periodic code sequence,

k is the range delay of the clutter source in units of clock period,

$g_o = \sqrt{\frac{m^2}{1+m^2}}$; g_o^2 is the DC component of the clutter power,

m^2 is the clutter DC-to-AC power ratio,

$g_a = \sqrt{\frac{1}{2(1+m^2)}}$; $2g_a^2$ is the AC component of the clutter power,

ψ_c is a random phase angle which is constant over a look period,

and $\{f_I(i)\}$ and $\{f_Q(i)\}$ are samples of the AC clutter component. The random sequences $\{f_I(i)\}$ and $\{f_Q(i)\}$ are assumed to have a Gaussian distribution with zero mean and unit variance and the two sequences are independent of each other. However, each sequence is assumed to be highly correlated and to have a Gaussian spectrum with zero mean and a standard deviation of σ_c Hz. Samples of the random clutter component are generated by introducing the desired correlation to an independent random sequence. This is accomplished using a linear system, and the process is explained in Appendix G.

Various types of clutter may be produced by proper use of the above procedure. Pure DC clutter resulting from static reflectors such as buildings can be obtained by setting g_a to zero and g_o to one. Antenna spillover can be simulated as a DC clutter with zero range delay, that is, k=0. Distributed clutter is generated by adding the clutter returns from several range cells. The return from each range cell must take into account the power variation according to the inverse of the fourth power of range, and the delay corresponding to the particular cell.

6.3 PROCESSOR SIMULATION

Since the proposed digital processors contain similar components but in different orders, each of the blocks was seperately simulated. This includes A/D converter, decoder, pulse compression filter, delay line canceller, weighting function, digital filter for clutter rejection and sampling rate reduction, FFT processor, and non-coherent integrator. Simulation of the digital processors is not complete in that only the quantization errors due to A/D conversion can be simulated. Finite word length effects elsewhere in the processor including the fast Fourier transformation have not been included. The notch and/or sampling rate reduction filter is assumed to be a recursive filter and is implemented as a cascade of second and first order sections to minimize the effects of coefficient quantization. Each of the first and second order sections is realized in the canonical structure. It should be noted that processor Configuration D-9 cannot be simulated with the programs that have been developed because of its many differences in comparision with the other configurations.

6.4 A SIMULATION EXPERIMENT

The simulations developed were exercised for a typical set of parameters. The object of the experiment was twofold: (i) to study the feasibility of digital processing when the target return is much smaller than the quantization interval of the A/D converter and (ii) to compare the performance of a digital processor with that of the analog processor described in Chapter 3.

Configuration D-8 was chosen as the candidate digital processor. Figure 6.1 shows the magnitude response of a fourth order Chebyshev low pass filter used for sampling rate reduction. This filter has 5 dB ripples in its pass band but, using the same argument as with the delay line canceller, this should not affect the output signal-to-noise ratio due to the narrow doppler filters provided by FFT processing. The output of this low pass filter was resampled at 80 KHz as opposed to 64 KHz suggested

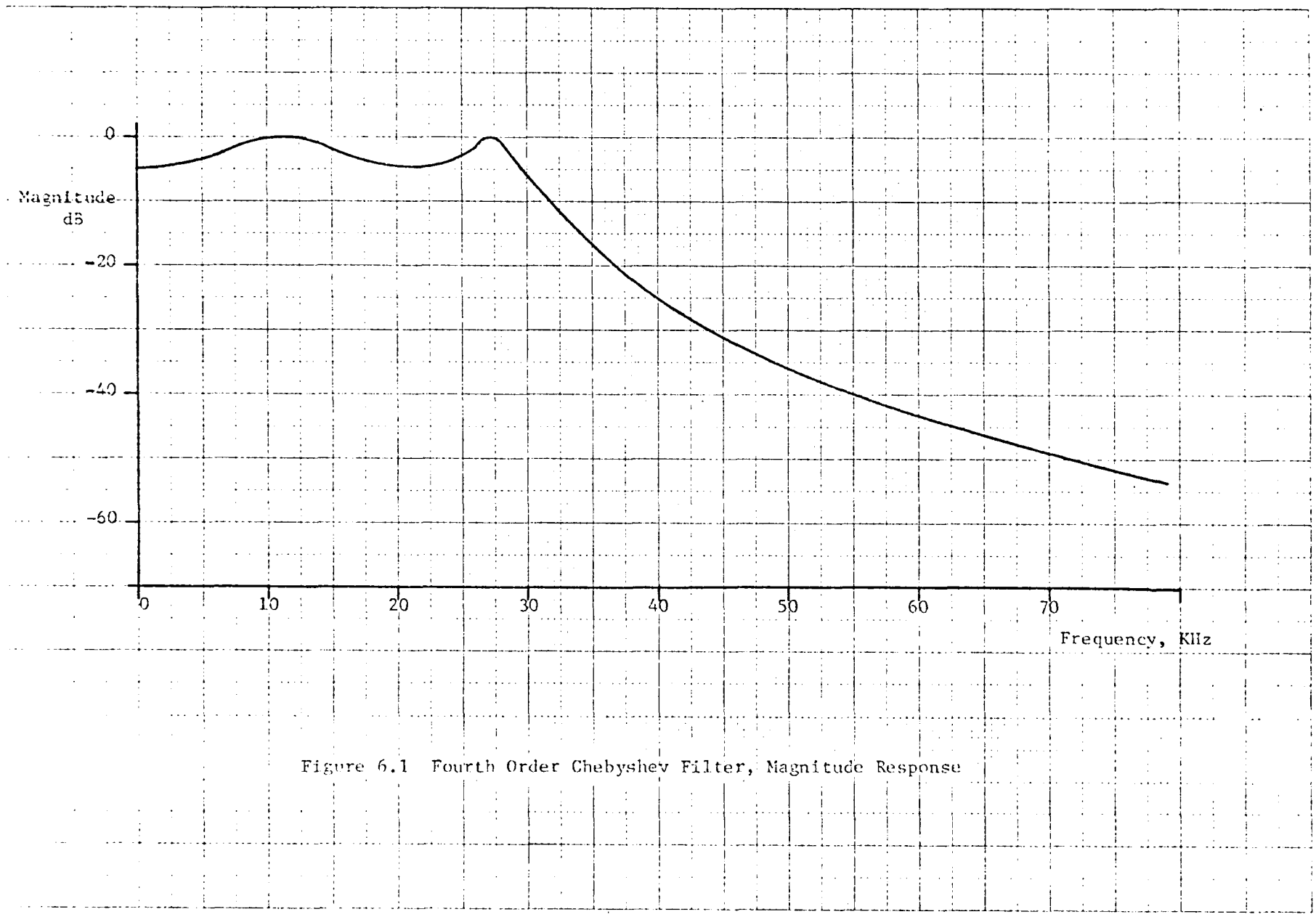


Figure 6.1 Fourth Order Chebyshev Filter, Magnitude Response

in Figure 5.9. The input samples to the FFT processor were weighted with a Hamming function shown in Figure 6.2. The simulation of the analog processor is exactly similar to Configuration D-1 with one difference. The non-coherent integration is performed at the original sampling rate instead of a reduced sampling rate to provide a better simulation of the analog integration.

The analog notch/low pass filter was modelled by a 10th order Butterworth low pass filter in cascade with an 8th order Butterworth high pass filter. The magnitude response, shown in Figure 6.3, was found to be a satisfactory approximation to the response of the analog filter. Assuming a look time of 2 milliseconds, a non-symmetrical Taylor weighting function (with $\bar{n}$ = 10 and -50 dB sidelobes), shown in Figure 6.4, was applied to the input prior to filtering. As explained in Chapter 3, the transient response must be allowed to settle down before starting integration. In this experiment, the non-coherent integration was started after 1.33 milliseconds, i.e., out of the 2 milliseconds look time, integration was performed only over 0.67 millisecond.

Table 6.1 shows the radar parameters used in the simulation. For these set of typical radar parameters, the signal-to-noise ratio at the input to the receiver can be computed to be -20 dB using the radar range equation. A complete list of other simulation parameters is given in Table 6.2.

6.5 SIMULATION RESULTS

The outputs in range channels 32, 28 and 36 were computed for both the analog and the digital processor using the parameters given above. These channels represent the in-range and two out-of-range channels all of which contain clutter and noise. A Monte-Carlo analysis of 30 samples was performed to determine the means and variances of the outputs. These results are given in Table 6.3 where the output

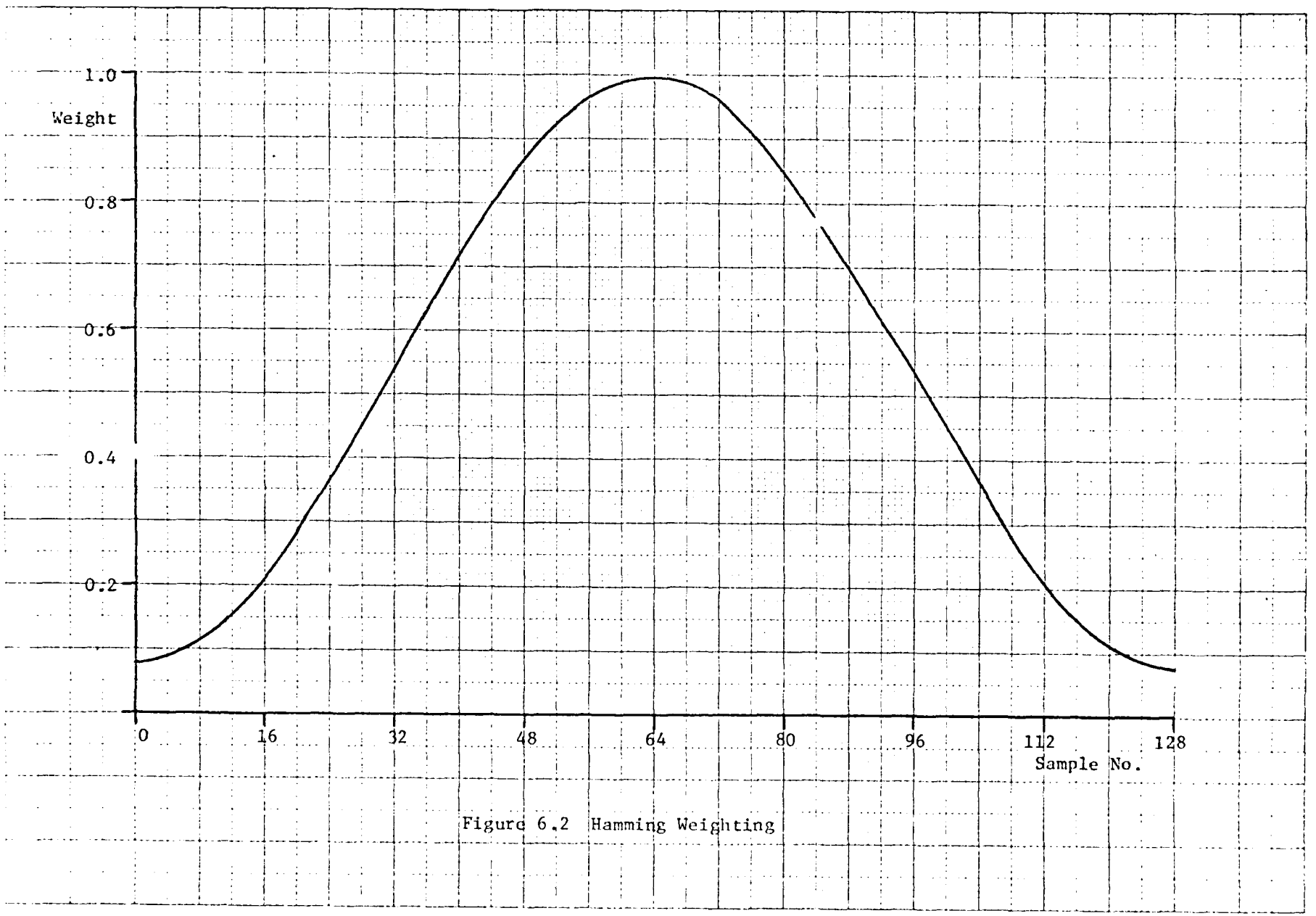


Figure 6.2 Hamming Weighting

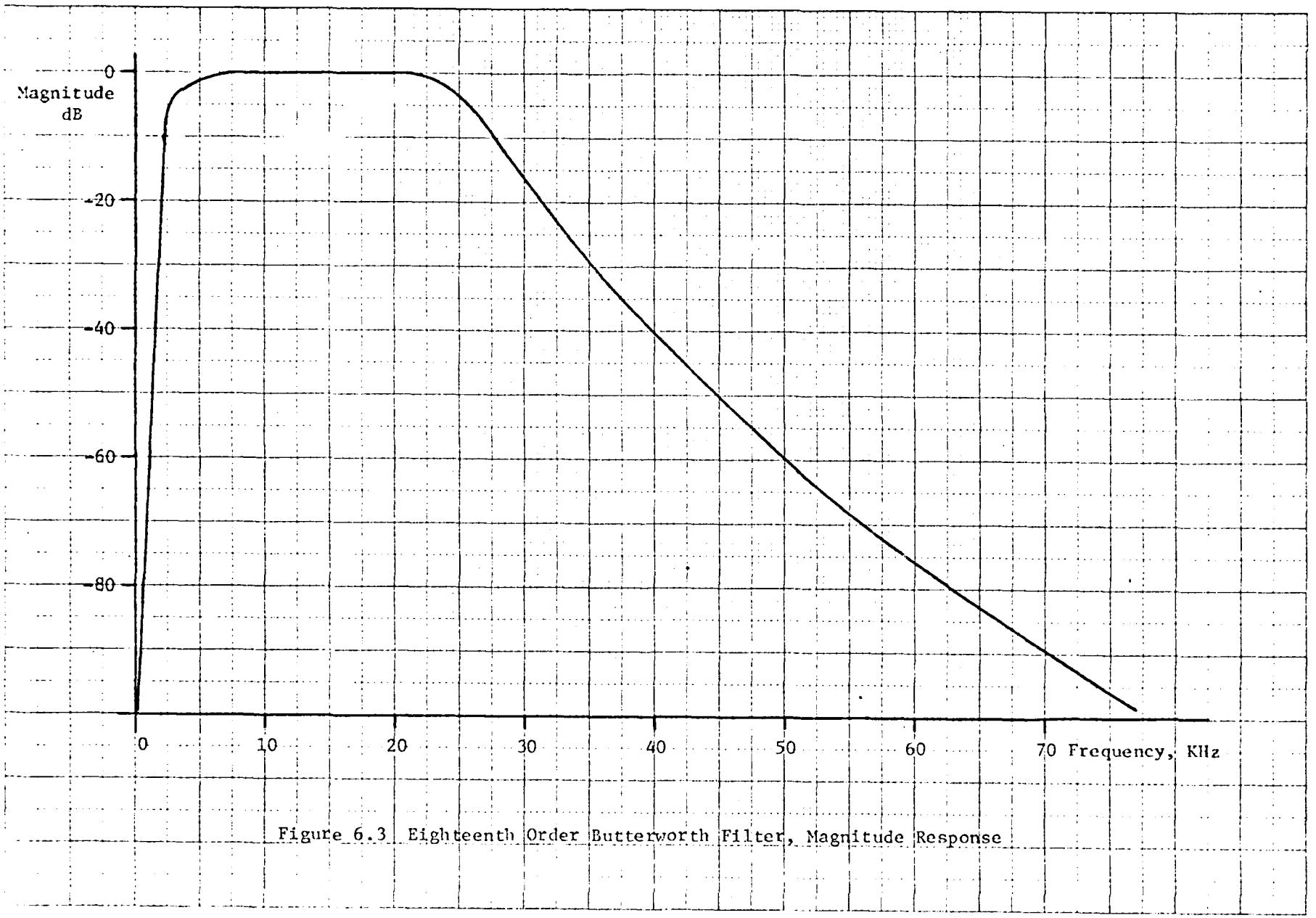


Figure 6.3 Eighteenth Order Butterworth Filter, Magnitude Response

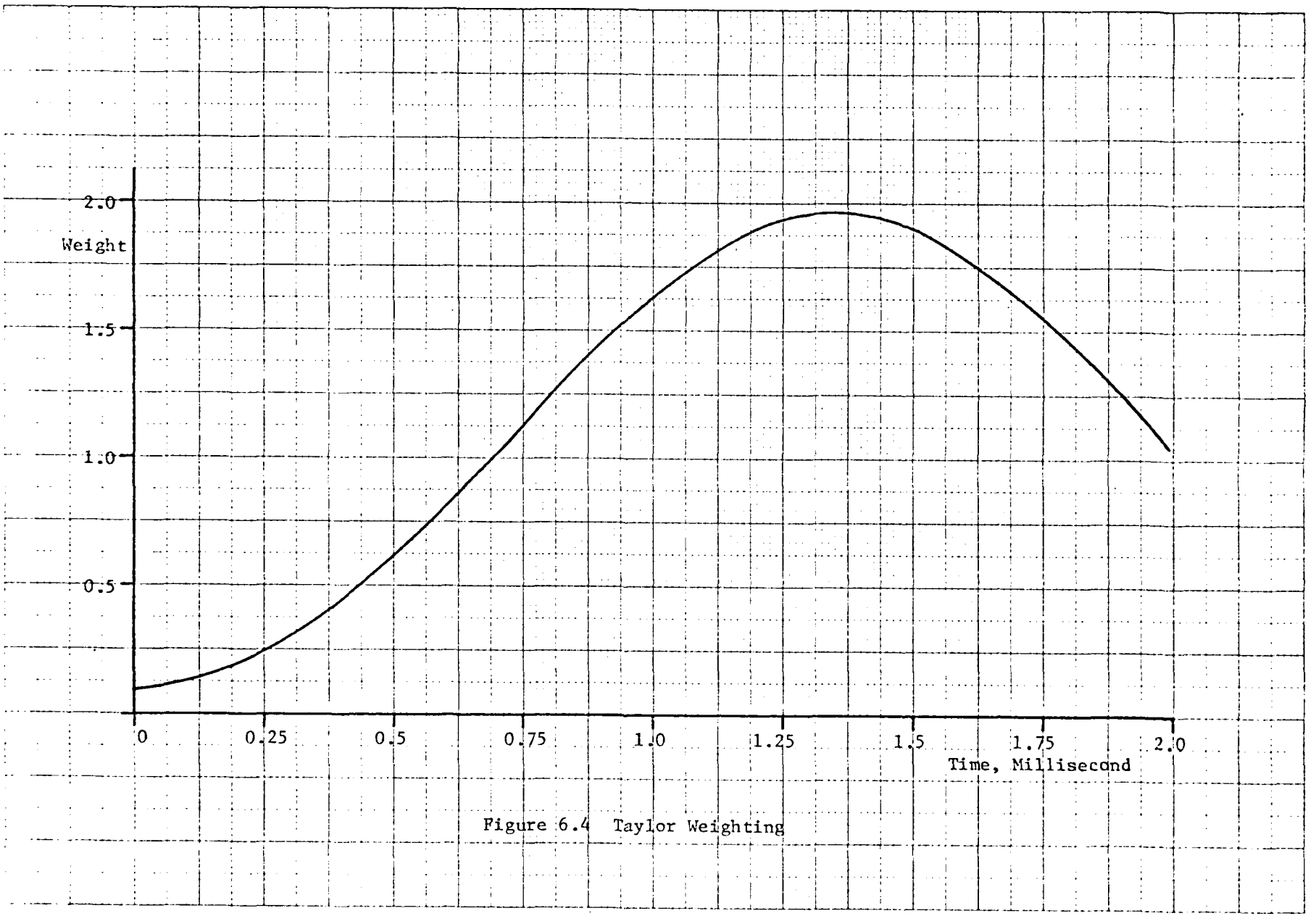


Figure 6.4 Taylor Weighting

Table 6.1 Radar Parameters Used in Simulation

Transmitted Power	10 watts
Transmit Antenna Gain	38 dB
Receive Antenna Gain	40 dB
Wavelength (X-band)	0.03 meter
Target Cross Section (Steady)	1 sq. meter
Target Range	15000 meters
Length of PN Sequence	63
Clock Rate	5 MHz
Code Period	12.6 μsec.
Target Range Rate	240 meters/sec.
Doppler Frequency	16 KHz
Receiver Bandwidth	5 MHz
Noise Figure	5 dB
Losses	9.5 dB

Table 6.2 Simulation Parameters

Target:	1 sq. meter steady target in range cell 32
In-range Clutter:	Signal-to-clutter ratio = -70 dB DC-to-AC power ratio = 0.8
Out-of-range Clutter:	Signal-to-clutter ratio = -70 dB DC-to-AC power ratio = 0.8 Appears in range cell 28
Fixed Clutter:	Signal-to-clutter ratio = -60 dB Appears in range cell 32
Clutter Parameters:	Wooded Terrain Wind Velocity = 18 knots Clutter doppler spread σ_F = 8 Hz
Noise:	Signal-to-noise ratio = -20 dB
Sampling Rate:	5 MHz
A/D Converter Input Range:	-18000 to 18000 volts
Look Time:	2 milliseconds for analog processor 1.65 milliseconds for digital processor
Integration:	Square law detection and integration for 0.67 millisecond in analog processor 128 sample FFT in digital processor followed by a square law detector

Table 6.3 Simulation Results

		Range Bin 28	Range Bin 32	Range Bin 36
Analog Processor	Mean	8692.04	17612.51	8472.89
	Variance	4.6585×10^6	6.5340×10^6	1.4319×10^6
Digital Processor 36 Bits	Mean	0.5044	33.0759	0.4370
	Variance	0.1752	29.5580	0.1726
Digital Processor 10 Bits	Mean	1.2152	32.4057	0.9593
	Variance	2.9316	91.9885	0.5703
Digital Processor 13 Bits	Mean	0.5493	33.3664	0.4443
	Variance	0.1910	29.4330	0.2020

of the digital processor corresponds to the output in the doppler channel in which the target is present. The digital processor was analyzed for three different A/D converter word lengths: 36 bits, 10 bits and 13 bits. The 36 bit word length was included for validation purposes since the quantization effects at this word length are inconsequential.

6.6 ANALYSIS OF THE RESULTS

In the case of the digital processors, the mean of the output is the variance before detection since a square law detector is being used. Since the original bandwidth is 5 MHz and the detection bandwidth is 620 Hz (based on the look time), the expected gain in signal-to-noise ratio is 39 dB. Thus the expected signal-to-noise ratio at the output without quantization effects is 19 dB. Using the results for 36 bits and range bins 28 and 32, it can be shown that the signal-to-interference ratio is 18 dB. This agrees with the predicted value and suggests that the large amount of input clutter has an insignificant effect on the output.

With a 13 bit A/D converter, the peak signal amplitude is four times smaller than the quantization interval corresponding to a signal-to-quantization error ratio of -5 dB. The effect of this is to decrease the input signal-to-noise ratio from -20 dB to -20.13 dB. The results show that the output signal-to-noise ratio is 17.75 dB, which is a fraction of a dB below that for the case where the word length is 36 bits. It should be noted that with 13 bits, the standard deviation of the input noise is about twice the quantization interval.

When the word length is reduced to 10 bits, the input noise standard deviation is one-fifth the quantization interval. The corresponding signal-to-quantization error is -23.15 dB which has the effect of increasing the input signal-to-noise ratio from -20 dB to -24.7 dB. From the results it can be shown that the output signal-to-noise ratio is 14 dB which is 4 dB less than the unquantized case.

Due to the noncoherent integration and the fact that clutter residues persist despite the weighting and delaying the start of integration, it is not straightforward to interpret the signal-to-interference ratio at the output of the analog processor. The most desirable way to analyze this processor is to perform an extensive Monte Carlo analysis to ascertain the probabilities of detection and false alarm. Such a simulation requires an enormous amount of computation. For purposes of this preliminary comparison, probabilities of detection and false alarm were computed by using approximate probability density functions. Probability densities with and without a target were approximated by a non-central and a central chi-squared distribution respectively. The appropriate parameters of these distributions were derived using the results of the simulation. Then, by varying the threshold, a curve of probability of false alarm versus probability of detection was developed for each processor. This set of curves is shown in Figure 6.5.

6.5 CONCLUSIONS

The results of the simulation show that it is indeed possible to extract the signal using a digital processor whose word length is such that the signal is considerably smaller than a quantization interval. Furthermore, detection is possible even when the input noise standard deviation is smaller than the quantization interval. Despite the large transients caused by clutter, it is shown that the analog processor can be used if a proper weighting function is used. However, for the set of parameters used, a look time of 2 milliseconds does not seem to provide an adequate detection performance by the analog processor. Finally, for a given look time, the digital processor gives a considerably better detection performance. This is attributable to the absence of undesriable transients and the use of coherent integration.

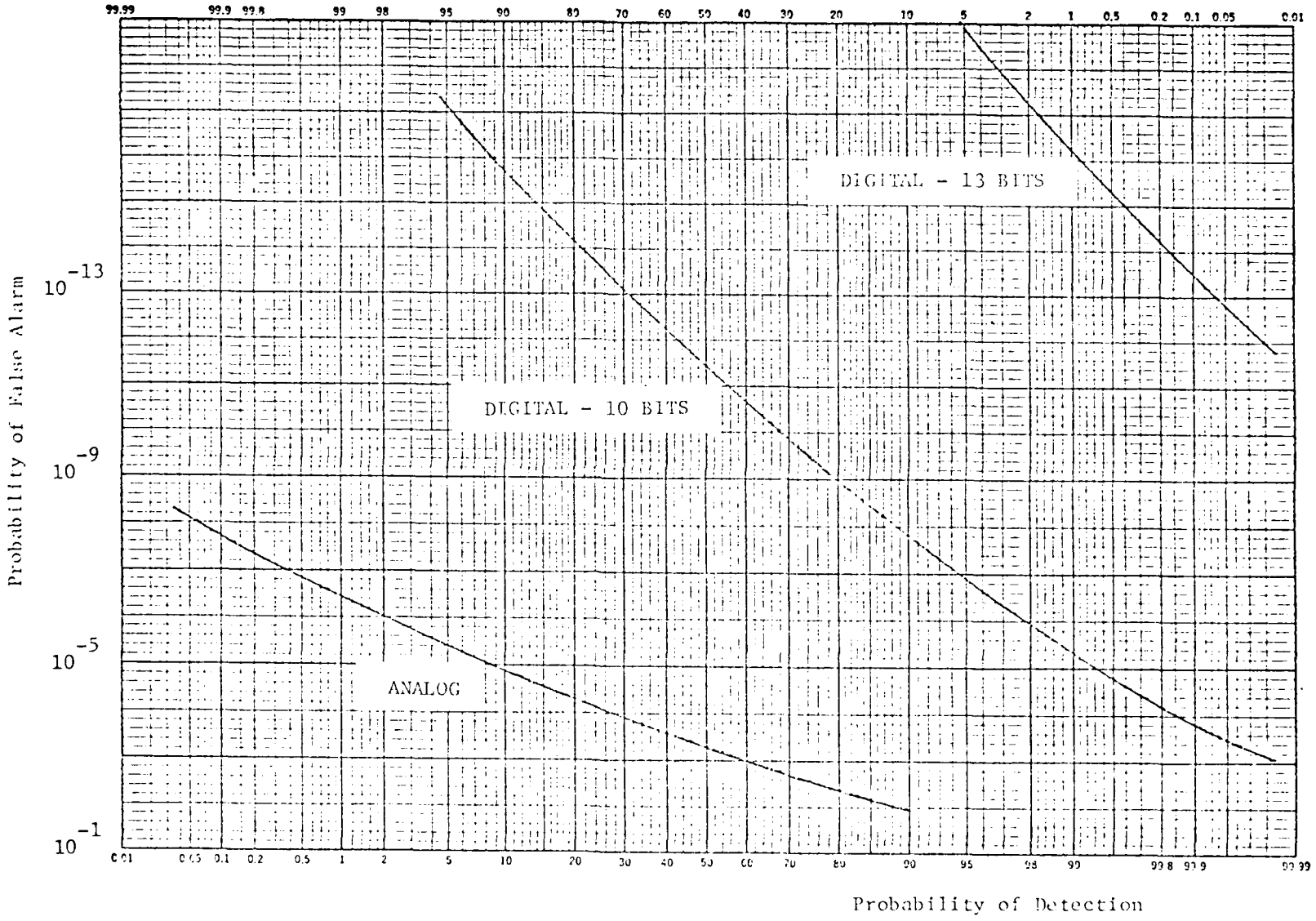


Figure 6.5 Performance Comparison

SECTION 7 - SUMMARY, CONCLUSIONS, AND FUTURE EFFORTS

The principal problem that must be addressed in designing a signal processor for a phase coded CW radar is the large dynamic range requirement that is caused by antenna spillover and close-in clutter. Since the clutter return could be 60-80 dB over the target return, the clutter rejection filter must be designed to have a very short settling time. Otherwise, the transient response due to the large clutter return will have the effect of reducing the signal-to-interference ratio thereby deteriorating the detection performance of the radar. Since digital clutter rejection filters, such as delay line cancellers, can be designed to have very short settling times, the possibility of using digital processing must be given serious consideration. Another attractive feature of digital processors is that they are considerably more flexible than analog processors. The main disadvantage of digital processing is the introduction of quantization errors due to the finite word length.

For a given dynamic range requirement, quantization errors can be reduced by increasing the word length. But the high sampling rate made necessary by the use of a wideband signal imposes a limit on the word length that can be obtained from currently available analog-to-digital converters. At the dynamic range and sampling rate of interest in this application, the largest word length available has a quantization interval which is much bigger than the amplitude of the target return. This problem can be partially alleviated by using hybrid processors where some flexibility of digital processing is sacrificed to relax the stringent and conflicting requirements of large dynamic range and high sampling rate.

Several analog, digital and hybrid configurations were proposed in this report as candidates for signal processors in a binary phase coded CW radar. Synthetic

input signals including returns due to clutter, target and noise were used along with digital computer simulations of the processors to show the feasibility of digital processing. It was shown that is was possible to detect a target when the target return is much smaller than the quantization interval of the A/D converter. The target was detectable even when the input noise was smaller than the quantization interval. It was also shown that the analog processor could be used if the transient response of the clutter rejection filter were controlled by modifying the input signal by a suitable weighting function.

It must be remarked that this study is incomplete and that the most significant conclusion is that digital processing is feasible despite the rigid sampling rate and dynamic range requirements. The following is a partial list of topics that must be investigated before selecting the optimal processors.

1. Effects of arithmetic roundoff and quantization of filter coefficients should be studied. Since no adequate theoretical tools are available, this must be performed through computer simulation.

2. Each of the configurations should be analyzed to determine dynamic range requirements at different stages of the processor. This is required in order to be able to use the available word length in the most efficient manner.

3. The simulation programs must be modified for use in the SEL computer and/or the AP 120B array processor. This will facilitate the computation of detection and false alarm probabilities using an exhaustive Monte-Carlo analysis.

4. The feasibility of using a fixed window transveral filter for sampling rate reduction must be investigated. This is important since such a

filter can be implemented with much less hardware than a recursive filter.

5. Adaptive thresholding by the use of constant false alarm rate (CFAR) processors should be examined. This technique is useful in avoiding false alarms in the presence of large unexpected interference sources.

6. MTI filters other than delay line cancellers can be used for clutter rejection. These can be of either recursive or non-recursive variety. In the latter case proper initialization is necessary to avoid transient effects.

7. The study must be expanded to include the effects of bandlimiting on the code and the effects of scanning modulation where a continuous scan is used instead of a step scan.

8. The processor must be modified to operate in clutter environments other than ground clutter. This entails the use of notch filters for rejection of weather clutter and chaff.

9. The typical set of parameters used in this study yields an unambiguous range of 1890 meters. One way to remove this ambiguity is to increase the length of the PN sequence.

   This, however, reduces the code repetition frequency which in turn decreases the separation between blind speeds produced by the clutter rejection filter. To avoid blind speeds, some method analogous to staggered PRF must be employed. The extension of staggered PRF to phase coded CW radars is not obvious and needs further investigation. The enlarged length of the sequence also increases the number of range channels

and hence the complexity of the processor. Therefore, alternate schemes for unambiguous range measurement must be examined.

10. The maximum length binary sequence does not seem to be ideally suited for the pulse compression approach. It is therefore necessary to study other types of waveform to find one which provides a satisfactory performance with the pulse compression processor. It must be remembered that the pulse compression processors are considerably easier to implement than those using decoders.

11. Even though some of the digital processor configurations are mathematically equivalent, their performances may not be identical because of the nonlinearity involved in quantization. It is therefore necessary to evaluate all the configurations to select a processor which offers the best compromise between performance and cost of implementation.

APPENDIX A

RESPONSE OF THE FFT PROCESSOR

Let the input to the processor be a sampled complex exponential,

$$x_n = \exp(2\pi j f n \Delta), \quad n=0, 1, \ldots, N-1 \tag{A.1}$$

where

f = frequency

N = number of input samples,

and Δ = sampling interval.

The frequency resolution of the FFT processor is the reciprocal of the input duration, i.e.,

$$F = 1/N\Delta , \tag{A.2}$$

and the kth output sample corresponds to a frequency kF. The frequency response of the kth frequency bin will be defined as the magnitude of the kth output sample as a function of the input frequency f. This output sample is given by

$$X_k(f) = \frac{1}{N} \sum_{n=0}^{N-1} x_n w^{nk} \tag{A.3}$$

where

$$w = \exp(-2\pi j/N) \tag{A.4}$$

Using (A.1) and (A.2), (A.3) reduces to

$$X_k(f) = \frac{1}{N} \sum_{n=0}^{N-1} \exp(2\pi j n \Delta f_r) \tag{A.5}$$

where

$$f_r = f - k F. \tag{A.6}$$

The summation in (A.5) is a finite geometric series and can be written as

$$X_k(f) = \frac{\exp(2\pi j N \Delta f_r) - 1}{N \exp(2\pi j \Delta f_r) - 1} . \tag{A.7}$$

The above equation can be rearranged to give

$$\left| X_k(f) \right| = \frac{1}{N} \left| \frac{\sin \pi N \Delta f_r}{\sin \pi \Delta f_r} \right| . \tag{A.8}$$

For large values of N and small values of f_r, the frequency response can be approximated as,

$$\left| X_k(f) \right| \cong \text{sa}(\pi N \Delta f_r) \tag{A.9}$$

where the function sa(x) denotes sin x/x.

An examination of the above expression reveals the following properties:

(i) the response of the kth bin is centered at frequency kF,

(ii) the frequency response contains large sidelobes, and

(iii) there is considerable overlap between the responses of neighboring bins. In fact, at the frequency where the two responses cross, the gain is about 4 dB below the maximum gain.

If the input is an independent, zero mean noise sequence with variance 1, it can be easily shown that the output in each bin of the FFT processor is a zero mean noise whose variance is 1/N. Since the input signal-to-noise ratio is unity, the gain in signal-to-noise ratio is the same as the output signal-to-noise ratio which is,

$$SNR_o = \frac{|X_k(f)|^2}{1/N}$$

$$= N \left[sa\ (\pi\ N\ \Delta\ f_r)\right]^2. \qquad (A.10)$$

APPENDIX B

RESPONSE OF DELAY LINE CANCELLER

The input is assumed to be a finite duration signal,

$$r(t) = x(t)\, w_1(t) \qquad \text{(B-1)}$$

where $x(t)$ is an infinite duration signal and $w_1(t)$ is a window function given by

$$w_1(t) = 1 \quad \text{for } 0 < t < T$$

$$= 0 \quad \text{otherwise} . \qquad \text{(B.2)}$$

The output of the two pulse delay line canceller is

$$y(t) = x(t) - x(t-\Delta) \qquad \text{(B.3)}$$

and the impulse response of the filter is

$$h(t) = \delta(t) - \delta(t-\Delta) . \qquad \text{(B.4)}$$

The output can therefore be written as

$$y(t) = x(t)\, w_1(t) - x(t-\Delta)\, w_1\,(t-\Delta) \qquad \text{(B.5)}$$

and exists in the interval $0 < t < T + \Delta$.

To avoid the effects of transients, a portion of this signal is discarded. Specifically, two segments of $y(t)$ in the intervals $0 < t < \Delta$ and $T < t < T + \Delta$ are not

used in further processing. This is identical to multiplying y(t) by a window function $w_2(t)$ such that,

$$w_2(t) = 1 \quad \text{for } \Delta<t<T$$
$$= 0 \quad \text{otherwise} \cdot \tag{B.6}$$

But from the definitions of $w_1(t)$ and $w_2(t)$, it can be easily seen that

$$w_1(t)\, w_2(t) = w_2(t), \tag{B.7}$$

and $$w_1(t-\Delta)\, w_2(t) = w_2(t) \tag{B.8}$$

Therefore the final output is

$$z(t) = [x(t) - x(t-\Delta)]\, w_2(t) \tag{B.9}$$

Thus, the effect of discarding the transients is equivalent to passing the infinite duration signal through the canceller followed by windowing. The importance of this is in regard to ground clutter. If the transients are not discarded, the effect of finite observation time is to smear the spectrum of the input signal. Therefore, the low frequency ground clutter spectrum expands into the passband of the filter. Discarding the transients improves clutter rejection performance since spectrum smearing takes place after the filtering which is designed to attenuate the low frequency clutter components and, in particular, to suppress the DC clutter completely. It must be noted that this analysis is valid only for step scan systems since the input amplitude has been assumed constant. The effect of discarding transients is not so great in continuously scanned systems since the antenna pattern automatically produces a spectral widening of the input signal.

APPENDIX C

COHERENT INTEGRATION FOLLOWING MTI

The input sampled signal is assumed to be of the form,

$$r(n) = x(n) + p(n), \quad n = -1, 0, 1, \ldots, N-1 \tag{C.1}$$

where $x(n) = \exp(2\pi j\ f\ n\Delta)$ (C.2)

and $p(n)$ is a zero mean, independent noise sequence with unit variance. Notice that the input signal-to-noise ratio is unity. It is assumed that this signal is passed through a two pulse canceller followed by an FFT processor. The signal component of the output of the two pulse canceller after discarding transients is,

$$\begin{aligned} y(n) &= x(n) - x(n-1) \\ &= \exp(2\pi j\ f\ n\Delta)\ [1-\exp(-2\pi j\ f\Delta)] \end{aligned} \tag{C.3}$$

If this sequence is passed through an FFT processor, the kth output sample is

$$Y(k) = \frac{1}{N} \sum_{n=0}^{N-1} y(n) w^{nk} \tag{C.4}$$

where $w = \exp(-2\pi j/N)$.

From (C.3) and (C.4), one can write

$$\begin{aligned} Y(k) &= [1-\exp(-2\pi j\ f\Delta)]\ \frac{1}{N} \sum_{n=0}^{N-1} \exp(2\pi j\ f\ n\Delta) w^{nk} \\ &= [1-\exp(-2\pi j f\Delta)]\ \frac{1}{N} \sum_{n=0}^{N-1} \exp(2\pi j\ n\ \Delta f_r) \end{aligned} \tag{C.5}$$

where $f_r = f - kF$ (C.6)

and $F = 1/N\Delta$. (C.7)

It follows that,

$$\left| Y(k) \right| = \left| 2 \sin \pi f\Delta \right| \left| \frac{\sin \pi N \Delta f_r}{N \sin \pi \Delta f_r} \right| . \quad (C.8)$$

The noise samples at the output of the canceller are

$$m(n) = p(n) - p(n-1), \; n=0, 1, \ldots, N-1. \quad (C.9)$$

Obviously, this is a zero mean noise sequence but is not an independent sequence due to the correlation introduced by the canceller. In fact, it can be easily shown that

$$\begin{aligned} E[m(n)m(i)] &= 2 \quad \text{if } n = i \\ &= -1 \quad \text{if } \left| n-i \right| = 1 \\ &= 0 \quad \text{otherwise} \end{aligned} \quad (C.10)$$

where E[.] denotes statistical expectation.

The kth output noise sample of the FFT processor is,

$$M(k) = \frac{1}{N} \sum_{n=0}^{N-1} m(n) w^{nk} . \quad (C.11)$$

This noise sample has zero mean and a variance given by,

$$E[\,|M(k)|^2] = \frac{1}{N^2} E \left[\sum_{n=0}^{N-1} \sum_{i=0}^{N-1} m(n)m(i) w^{kn} w^{-ki} \right] . \quad (C.12)$$

Using (C.10), the above equation can be reduced to

$$E[\,|M(k)|^2] = \frac{1}{N^2} E\left[\sum_{n=0}^{N-1} m^2(n)+m(n)m(n-1)w^{-k}+m(n)m(n-1)w^{k}\right]$$

$$= \frac{1}{N^2} \left[2N - N(w^k + w^{-k})\right]$$

$$= 4 \sin^2(\pi k F\Delta)/N. \qquad (C.13)$$

The total gain in signal-to-noise ratio, which is the same as the output signal-to-noise ratio is,

$$SNR_o = \frac{|Y(k)|^2}{E[|M(k)|^2]}$$

$$= \frac{4 \sin^2 \pi f \Delta}{4 \sin^2(\pi k F\Delta)/N} \left[\frac{\sin \pi N\Delta f_r}{N \sin \pi \Delta f_r}\right]^2 \qquad (C.14)$$

In the doppler bin containing the target, $f \cong kF$ and for large values of N, (C.14) can be approximated as,

$$SNR_o \cong N \left[sa(\pi N \Delta f_r)\right]^2 \qquad (C.15)$$

which is the same as that obtained by coherent integration without the delay line canceller. Thus, introducing a two pulse canceller ahead of the FFT processor has little effect on the improvement in signal-to-noise ratio. However, the signal levels in different frequency bins are not the same and each bin must be assigned a different threshold which takes into account the frequency response of the delay line canceller.

APPENDIX D

MTI BEFORE AND AFTER DECODING

The received signal consists of K periods of the code and there are N samples in each code period. These samples are denoted as $r(i)$, $i=0, 1, \ldots, KN-1$. Let this set be partitioned into K segments of N samples (one code period) each such that,

$$r_k(n) = r(kN + n) \qquad \text{for } k = 0, 1, \ldots, K-1,$$

$$\text{and } n = 0, 1, \ldots, N-1. \tag{D.1}$$

The output of the decoder in range channel corresponding to a delay of m samples is obtained by multiplying $r(i)$ by a periodic repetition of $C_m(n)$ where $C_m(.)$ represents a period of the code shifted by m samples. Thus, if the output $y(i)$ is also partitioned as above, then

$$y_k(n) = r_k(n)\, C_m(n) \; . \tag{D.2}$$

If this sequence is then passed through a two pulse canceller whose delay is an integer multiple of the code period, i.e., the delay is pk samples, the output of the canceller in partitioned form is

$$z_k(n) = y_k(n) - y_{k-p}(n)$$

$$= [r_k(n) - r_{k-p}(n)]\, C_m(n) \; . \tag{D.3}$$

If the MTI is performed before decoding, the output of the two pulse canceller in partitioned form is

$$x_k(n) = r_k(n) - r_{k-p}(n) \; . \tag{D.4}$$

If the sequence $x_k(n)$ is then decoded, the decoder output in the range channel corresponding to a delay of m samples is,

$$v_k(n) = x_k(n) \cdot C_m(n)$$

$$= [r_k(n) - r_{k-p}(n)]\, C_m(n) \cdot \qquad \text{(D.5)}$$

From (D.3) and (D.5) it is seen that interchanging the MTI and decoding operations has no effect on the output. However, it must be noted that this result is based on the assumption that the canceller delay is an integer multiple of the code period. The result is not valid for other delays and for recursive MTI filters.

APPENDIX E

OUTPUT OF PROCESSORS USING DECODERS

In order to demonstrate the equivalence of Configurations D-5 and D-9, expressions for their outputs are developed in this Appendix. In the development that follows, the delay line canceller is ignored since its effects on the two configurations are identical. The input signal is denoted by the set of samples $\{r(i),\ i = 0, 1, \ldots, KN-1\}$ where K is the number of code periods in the look time and N is the number of samples in each code period.

In Configuration D-5, the output of the range channel corresponding to a delay of m samples is obtained by multiplying the input by a periodic repetition of $C_m(n)$ where $C_m(.)$ represents a period of the code shifted by m samples. Thus, the decoder output is given by,

$$y_m(i) = r(i) \sum_{k=0}^{K-1} C_m(i-kN) \qquad (E.1)$$

The nth output sample following an NK sample FFT is

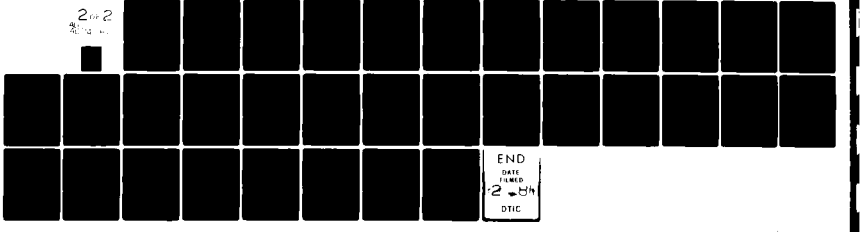
$$Y_m(n) = \sum_{i=0}^{NK-1} y_m(i) w^{in} \qquad (E.2)$$

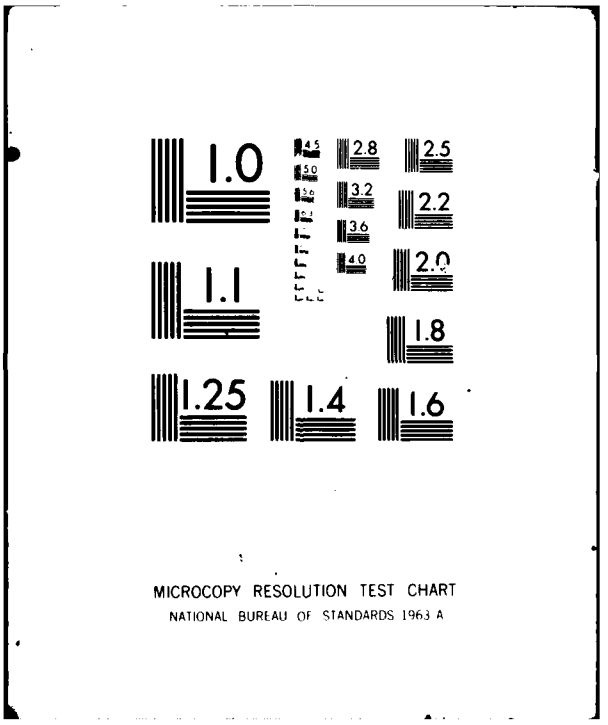
where $w = \exp(-2\pi j/NK)$. (E.3)

Combining (E.1) and (E.2),

$$Y_m(n) = \sum_{i=0}^{NK-1} \sum_{k=0}^{K-1} r(i)\ C_m(i-kN) w^{in} \qquad (E.4)$$

AD-A094 181 COMPUTER SCIENCES CORP HUNTSVILLE AL F/G 17/9
SIGNAL PROCESSOR FOR BINARY PHASE CODED CW RADAR,(U)
DEC 77 B K BHAGAVAN DAAH03-75-A-0045
UNCLASSIFIED CSC/TR-77/5491 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Each of the K sequences $\{U(p,n),\ p = 0, 1, \ldots, N-1\}$ is doppler compensated, i.e.,

$$X(p,n) = U(p,n)\ w^{np} \tag{E.10}$$

The output of the processor in range cell m and doppler cell n is then computed by decoding followed by summation,

$$V_m(n) = \sum_{p=0}^{N-1} X(p,n)\ C_m(p) \tag{E.11}$$

$$= \sum_{p=0}^{N-1} Z_p(n)\ w^{np}\ C_m(p)$$

$$= \sum_{p=0}^{N-1} \sum_{k=0}^{K-1} z_p(k) w_1^{nk}\ w^{np}\ C_m(p)$$

$$= \sum_{p=0}^{N-1} \sum_{k=0}^{K-1} z_p(k)\ w^{nkN}\ w^{np}\ C_m(p)$$

$$= \sum_{p=0}^{N-1} \sum_{k=0}^{K-1} r(kN+p)\ C_m(p)\ w^{np}\ w^{nkN}\ . \tag{E.12}$$

Comparison of (E.5) and (E.12) proves the hypothesis that the outputs of Configurations (D-5) and (D-9) are identical.

APPENDIX F

OUTPUT OF PROCESSORS USING PULSE COMPRESSION

Referring to Configuration PC-2 and ignoring the delay line canceller as in Appendix E, the output of the pulse compression filter in range channel m is given by

$$y_m(k) = \sum_{p=0}^{N-1} C_m(p) r(kN+p) \quad . \tag{F.1}$$

If the K sample FFT of this sequence is computed, the output sample in the nth bin is

$$Y_m(n) = \sum_{k=0}^{K-1} y_m(k)\ w_1^{nk} \tag{F.2}$$

Here $w_1 = \exp(-2\pi j/K)$

$$= w^N \tag{F.3}$$

where

$$w = \exp(-2\pi j/NK) \tag{F.4}$$

Therefore,

$$Y_m(n) = \sum_{k=0}^{K-1} \sum_{p=0}^{N-1} r(kN+p)\ C_m(p) w^{nkN} \quad . \tag{F.5}$$

Comparison of (F.5) and (E.12) reveals the similarity between the decoder and the pulse compression approaches, the only difference being the term due to doppler compensation.

APPENDIX G

CLUTTER GENERATION

It is desired to generate samples of the random clutter components $\{f_I(i)\}$ and $\{f_Q(i)\}$ at sampling intervals of δ. It is assumed that the sequence can be described by a Gaussian distribution with zero mean and unit variance. These sequences are required to be highly correlated having Gaussian power spectrum centered at zero frequency with a standard deviation of σ_F. Thus, the desired power spectrum is

$$\Phi_{ff}(f) = \frac{1}{\sqrt{2\pi}\ \sigma_F} \exp(-f^2/2\sigma_F{}^2) \ . \tag{G.1}$$

Consider a continuous linear system whose impulse response is

$$g(t) = \exp(-t^2/\sigma_\tau^2) \tag{G.2}$$

where

$$\sigma_\tau = 1/2\pi\sigma_F \ . \tag{G.3}$$

The transfer function of this non-causal system is

$$\left|G(f)\right| = \frac{1}{2\sqrt{\pi}\ \sigma_F} \exp(-f^2/4\sigma_F^2) \ . \tag{G.4}$$

Let the input to this system, denoted n(t), be a zero mean, unit variance, Gaussian noise process with a flat spectrum,

$$\begin{aligned} \Phi_{nn}(f) &= S \text{ for } \left| f \right| < \frac{1}{2S} \\ &= 0 \quad \text{otherwise} \ . \end{aligned} \tag{G.5}$$

Then, the output process c(t) is also a zero mean Gaussian process whose power spectrum is

$$\begin{aligned}\Phi_{cc}(f) &= \left|G(f)\right|^2 \Phi_{nn}(f) \\ &= \frac{S}{4\pi\sigma_F^2} \exp(-f^2/2\sigma_F^2) \qquad \text{for } |f| < \frac{1}{2S} \\ &= 0 \qquad \text{otherwise} \end{aligned} \tag{G.6}$$

The restriction on the region of existence can be removed if S is chosen to be small enough such that

$$\frac{1}{2S} > 2\sigma_F \ . \tag{G.7}$$

From its power spectrum, it is obvious that if the input noise is sampled at intervals of S, the samples are independent and have a Gaussian distribution with zero mean and unit variance. Thus, a discrete equivalent of the continuous system may be written as,

$$c(t) = S \sum_k n(kS)\ g(t-kS) \tag{G.8}$$

If c(t) is sampled at intervals of δ,

$$\begin{aligned} c(m\delta) &= S \sum_k n(kS)\ g(m\delta - kS) \\ &= S.\ f(m) \ . \end{aligned} \tag{G.9}$$

The power spectrum of {f(m)} is,

$$\Phi_{ff}(f) = \frac{1}{S^2}\,\Phi_{cc}(f)$$

$$= \frac{1}{S.\ 4\pi\sigma_F^2}\exp(-f^2/2\sigma_F^2)$$

$$= \frac{1}{2\sqrt{2\pi}\ \sigma_F}\exp(-f^2/2\sigma_F^2) \qquad \text{(G.10)}$$

where $S = \dfrac{1}{2\sqrt{2\pi}\ \sigma_F}$ (G.11)

Note that the definition of S satisfies the inequality given in (G.7) and that the power spectrum of {f(m)} is the same as that of the required clutter components. Thus, the clutter sequence can be generated using

$$f(m) = \sum_k n(kS)\,g(m\delta - kS)\ . \qquad \text{(G.12)}$$

However, it must be noted that the impulse response of the filter has infinite duration. For implementation in a digital simulation, the impulse response is truncated to $|t|<3S$. Since the input noise sampled at intervals of S, six samples of noise are required in the summation in (G.12). Note however, that several samples of clutter can be generated from the same six noise samples because the output sampling interval δ is much smaller than S. In fact, I samples of clutter can be computed from six samples of input noise where $I = S/\delta$.

APPENDIX H

PROGRAM LISTINGS

This appendix contains the listings of all the computer programs developed for this study. The programs re in FORTRAN and are compatible for execution on the CDC-6600 computer system.

```
SUBROUTINE ANIT       74/74   OPT=1                    FTN 4.2+74355      1

      SUBROUTINE ANIT(START)
C*****************************************************************************
C**INITIALIZATION FOR NOISE GENERATORS
C*****************************************************************************
      COMMON/RANDM/ARD,RND1,RS
      ARD=START
      RND1=START*0.000001
      RS=47436.0
      RETURN
      END
```

```
      SUBROUTINE CLUTTN(X,SCR,XM2,SIGMAF,NCDEL)
C*****************************************************************
C**GENERATES SAMPLES OF GROUND CLUTTER AND ADDS
C**THEM TO THE COMPLEX ARRAY X
C**X IS THE INPUT/OUTPUT COMPLEX ARRAY
C**SCR IS THE SIGNAL TO CLUTTER RATIO
C**NOTE==SIGNAL POWER IS ASSUMED TO BE UNITY
C**XM2 IS THE DC TO AC POWER RATIO
C**AC CLUTTER HAS GAUSSIAN SPECTRUM
C**WITH ZERO MEAN
C**SIGMAF IS THE STANDARD DEVIATION OF
C**THE CLUTTER SPECTRUM
C**NCDEL IS THE RANGE DELAY CORRESPONDING
C**TO THE CLUTTER LOCATION
C*****************************************************************
      COMMON/ADPAR/DELT,QAD,VMAX
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMMON/XCODE/NSTAGE,INIT(10),FEEDBK(10),CODE(1024)
      COMPLEX X(1)
      DIMENSION XR(4),XN(12)
      INTEGER CODE
      CONST=1.0/(SQRT(SCR))
      NCYCLE=NLOOK/NSAMPP
      CALL RANDU(XR,2)
      PI=3.141492654
      PSI=2.0*PI*XR(2)
      CPSI=COS(PSI)
      SPSI=SIN(PSI)
      IF(XM2.LT.0) GO TO 20
      GO=SQRT(XM2/(1.0+XM2))
      GA=SQRT(0.5/(1.0+XM2))
      GC=GO*CPSI
      GS=GO*SPSI
      CALL RANDG(XN,12,0.0,1.0)
      PI2=2.0*PI
      SIGTAU=1.0/(PI2*SIGMAF)
      SIGSS=SIGTAU/DELT
      SIG1=SIGSS/1.414213562
      CDEN=0.5/(SIG1*SIG1)
      ITI=1.253314137*SIGSS
      TI=FLOAT(ITI)
      DO 10 IC=1,NCYCLE
      DO 11 ID=1,NSAMPP
      J=(IC-1)*NSAMPP+ID
      ICODE=ID-NCDEL
      IF(ICODE.LE.0)ICODE=ICODE+NSAMPP
      CI=0.0
      CQ=0.0
      DO 12 I=1,6
      M=(6-I)*ITI+J
      B=FLOAT(M)
      DUM=B-3.0*TI
      EXPT=-DUM*DUM*CDEN
      HM=EXP(EXPT)
      CI=CI+XN(I)*HM
   12 CQ=CQ+XN(I+6)*HM
      CIT=FLOAT(CODE(ICODE))*(GC+GA*CI)
```

```
      CQT=FLOAT(CODE(ICODE))*(GS+GA*CQ)
      X(J)=X(J)+ CONST*CMPLX(CIT,CQT)
   11 CONTINUE
   10 CONTINUE
      RETURN
   20 CONTINUE
      DO 30 IC=1,NCYCLE
      DO 31 ID=1,NSAMPP
      J=(IC-1)*NSAMPP+ID
      ICODE=ID-NCDEL
      IF(ICODE.LE.0)ICODE=ICODE+NSAMPP
      CIT=FLOAT(CODE(ICODE))*CPSI
      CQT=FLOAT(CODE(ICODE))*SPSI
      X(J)=X(J)+ CONST*CMPLX(CIT,CQT)
   31 CONTINUE
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CORREL(X,NIN,NOUT)
C*****************************************************************
C**PULSE COMPRESSION FILTER FOR THE
C**PROCESSORS USING PULSE COMPRESSION
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C**NIN IS THE NUMBER OF INPUT SAMPLES
C**NOUT IS THE NUMBER OF OUTPUT SAMPLES
C*****************************************************************
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMMON/XCODE/NSTAGE,INIT(10),FEEDBK(10),CODE(1024)
      COMPLEX X(1),XOUT
      INTEGER CODE
      NOUT=NIN+NSAMPP
      NSMPP1=NSAMPP+1
      NINP=NIN+1
      DO 10 II=NINP,NOUT
      I=NOUT-II+NINP
      XOUT=CMPLX(0.0,0.0)
      JFIN=NOUT-I
      DO 11 J=1,JFIN
      IND=J+I-NSAMPP
      IF(CODE(J).GT.0)GO TO 12
      XOUT=XOUT-X(IND)
      GO TO 11
   12 XOUT=XOUT+X(IND)
   11 CONTINUE
      X(I)=XOUT
   10 CONTINUE
      DO 20 II=NSMPP1,NIN
      I=NIN-II+NSMPP1
      XOUT=CMPLX(0.0,0.0)
      DO 21 J=1,NSAMPP
      IND=J+I-NSAMPP
      IF(CODE(J).GT.0)GO TO 22
      XOUT=XOUT-X(IND)
      GO TO 21
   22 XOUT=XOUT+X(IND)
   21 CONTINUE
      X(I)=XOUT
   20 CONTINUE
      DO 30 II=1,NSAMPP
      I=NSAMPP-II+1
      XOUT=CMPLX(0.0,0.0)
      JST=NSAMPP-I+1
      DO 31 J=JST,NSAMPP
      IND=J+I-NSAMPP
      IF(CODE(J).GT.0)GO TO 32
      XOUT=XOUT-X(IND)
      GO TO 31
   32 XOUT=XOUT+X(IND)
   31 CONTINUE
      X(I)=XOUT
   30 CONTINUE
      RETURN
      END
```

```
SUBROUTINE DECODE     74/74   OPT=1                    FTN 4.2+74355      1

      SUBROUTINE DECODE(X,N,IR)
C*************************************************************************
C**IMPLEMENTS THE DECODER
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C**N IS THE NUMBER OF SAMPLES
C**IR IS THE REQUIRED RANGE BIN
C*************************************************************************
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMMON/XCODE/NSTAGE,INIT(10),FEEDBK(10),CODE(1024)
      COMPLEX X(1)
      INTEGER CODE
      NCYCLE=N/NSAMPP
      DO 10 IC=1,NCYCLE
      DO 10 ID=1,NSAMPP
      J=(IC-1)*NSAMPP+ID
      ICODE=ID-IR
      IF(ICODE.LE.0)ICODE=ICODE+NSAMPP
      IF(CODE(ICODE).LT.0)X(J)=-X(J)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DFT(A,N,ISN,NBIN)
C*****************************************************************
C**COMPUTES THE DISCRETE FOURIER TRANSFORM OF
C**THE COMPLEX ARRAY X
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C**N IS THE NUMBER OF SAMPLES IN X
C**ISN = -1 FOR DIRECT DFT
C**ISN = +1 FOR INVERSE DFT
C**NBIN IS NUMBER OF OUTPUT SAMPLES DESIRED
C**IF N = 2**M , FFT IS USED
C**OTHERWISE DFT IS USED
C*****************************************************************
      COMPLEX A(1),T1,T2,TEMP,X(256)
      PI2=6.28318530717959
      MASK=1
      IGAM=0
    1 IGAM=IGAM+1
      M=2**IGAM
      IF(N-M)2,4,1
    4 CONTINUE
      N1=N-1
      DO 20 II=2,N1
      I=II-1
      IFLIP=0
      DO 10 J=1,IGAM
      LSN=2**(J-1)
   10 IFLIP=2*IFLIP+(MASK.AND.(I/LSN))
      IF(I.LE.IFLIP)GO TO 20
      I1=I+1
      I2=IFLIP+1
      TEMP=A(I2)
      A(I2)=A(I1)
      A(I1)=TEMP
   20 CONTINUE
      DO 30 I=1,IGAM
      NEL=2**I
      NEL2=NEL/2
      NSET=N/NEL
      ANG=PI2/NEL
      SI=SIN(ANG)
      CI=COS(ANG)
      DO 30 J=1,NSET
      INCR=(J-1)*NEL
      S0=0.0
      C0=1.0
      DO 30 II=1,NEL2
      J1=II+INCR
      J2=J1+NEL2
      T1=A(J1)
      T2=A(J2)*CMPLX(C0,ISN*S0)
      A(J1)=T1+T2
      A(J2)=T1-T2
      SN=S0*CI+C0*SI
      CS=C0*CI-S0*SI
      C0=CS
   30 S0=SN
      GO TO 100
```

```
      2  CONTINUE
         ANG=ISN*PI2/FLOAT(N)
         DO 40 I=1,NBIN
         X(I)=CMPLX(0.0,0.0)
         DO 50 J=1,N
         ARG=ANG*FLOAT((I-1)*(J-1))
         T1=CMPLX(0.0,ARG)
         T2=CEXP(T1)
         X(I)=X(I)+T2*A(J)
     50  CONTINUE
     40  CONTINUE
         DO 60 I=1,N
     60  A(I)=X(I)
    100  CONTINUE
         IF(ISN.GT.0)RETURN
         DO 110 I=1,N
         A(I)=A(I)/FLOAT(N)
    110  CONTINUE
         RETURN
         END
```

```
      SUBROUTINE FFTDEC(X,NSTART,NSKIP,NBIN,NOUT)
C*****************************************************************
C**FFT PROCESSING FOR THE DECODER APPROACH
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C**NSTART IS  SAMPLE WHERE INTEGRATIN BEGINS
C**NSKIP SAMPLING RATE REDUCTION FACTOR
C**NBIN IS THE NUMBER OF OUTPUT BINS DESIRED
C**NOUT = NBIN IN THIS SIMULATION
C*****************************************************************
      COMMON/PROPAR/NLOOK,NSAMPR,NFFT,NSCLK,NCINT
      COMPLEX X(1),XFFT(256)
      NS=1
      ISTIN=NSTART
      ISTOUT=0
      NOUT=NBIN*NCINT
      DO 10 INT=1,NCINT
      DO 11 IFFT=1,NFFT
      ISAMP=(IFFT-1)*NSKIP+ISTIN
      XFFT(IFFT)=X(ISAMP)
   11 CONTINUE
      CALL WEIGHT(XFFT,NFFT)
      CALL DFT(XFFT,NFFT,-1,NBIN)
      DO 12 IFFT=1,NBIN
      X(IFFT+ISTOUT)=XFFT(IFFT)
   12 CONTINUE
      ISTOUT=ISTOUT+NBIN
      ISTIN=ISTIN+NFFT*NSKIP
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FFTPC(X,NSTART,NBIN)
C*****************************************************************************
C**FFT PROCESSING FOR PULSE COMPRESSION
C**CONFIGURATIONS
C**X IS THE COMPLES INPUT ARRAY
C**NSTART SAMPLE AT WHICH INTEGRATION STARTS
C**NBIN IS THE NUMBER OF OUTPUT BINS DESIRED
C*****************************************************************************
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMPLEX X(1),XFFT(256)
      NCOH=NFFT*NSAMPP
      NFIN=NSTART+(NCOH*NCINT)-1
      DO 10 INT=NSTART,NFIN,NCOH
      DO 11 ISAMP=1,NSAMPP
      DO 12 IFFT=1,NFFT
      J=(INT-1)+(IFFT-1)*NSAMPP+ISAMP
      XFFT(IFFT)=X(J)
   12 CONTINUE
      CALL WEIGHT(XFFT,NFFT)
      CALL DFT(XFFT,NFFT,-1,NBIN)
      DO 13 IFFT=1,NFFT
      J=(INT-1)+(IFFT-1)*NSAMPP+ISAMP
      X(J)=XFFT(IFFT)
   13 CONTINUE
   11 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FILCAS(X,NSTART,NFIN)
C*****************************************************************************
C**IMPLEMENTS RECURSIVE DISCRETE FILTERS
C**AS CASCADE OF SECOND AND FIRST ORDER SECTIONS
C**CANONICAL REALIZATION IS USED
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C**NSTART IS SAMPLE WHERE FILTERING BEGINS
C**NFIN IS SAMPLE WHERE FILTERING ENDS
C*****************************************************************************
      COMMON/FLTT/NSEC,NFIRST,NTYPE(10),CSEC(2,10),CFIRST,CONST
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      LOGICAL NTYPE
      COMPLEX X(1),Y(3),DUM
      DO 10 ISEC=1,NSEC
      DO 11 I=1,3
   11 Y(I)=(0.0,0.0)
      DO 12 I=NSTART,NFIN
      Y(1)=X(I)
      DO 13 IF=1,2
   13 Y(1)=Y(1)-CSEC(IF,ISEC)*Y(IF+1)
      DUM=Y(1)+Y(3)
      IF(NTYPE(ISEC))GO TO 15
      DUM=DUM-Y(2)-Y(2)
      GO TO 16
   15 DUM=DUM+Y(2)+Y(2)
   16 CONTINUE
      X(I)=DUM
      Y(3)=Y(2)
      Y(2)=Y(1)
   12 CONTINUE
   10 CONTINUE
      IF(NFIRST.EQ.0)GO TO 25
      Y(1)=(0.0,0.0)
      Y(2)=(0.0,0.0)
      DO 22 I=NSTART,NFIN
      Y(1)=X(I)
      Y(1)=Y(1)-CFIRST*Y(2)
      IF(NTYPE(NSEC+1))GO TO 23
      DUM=Y(1)-Y(2)
      GO TO 24
   23 DUM=Y(1)+Y(2)
   24 CONTINUE
      X(I)=DUM
      Y(2)=Y(1)
   22 CONTINUE
   25 CONTINUE
      DO 30 I=NSTART,NFIN
   30 X(I)=X(I)*CONST
      RETURN
      END
```

```
SUBROUTINE INTDEC     74/74   OPT=1                    FTN 4.2+74355      1

      SUBROUTINE INTDEC(X,NBIN)
C*****************************************************************************
C**INTEGRATION FOR THE DECODER APPROACH
C*****************************************************************************
      COMMON/PROPAR/NLOOK,NSAMPR,NFFT,NSCLK,NCINT
      COMPLEX X(1)
      DO 10 IFFT=1,NBIN
      SUM=0.0
      DO 12 INT=1,NCINT
      ISAMP=(INT-1)*NBIN+IFFT
      SUM=SUM+CABS(X(ISAMP))**2
   12 CONTINUE
      X(IFFT)=CMPLX(SUM,0.0)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INTPC(X,NSTART)
C*****************************************************************
C**NON COHERENT INTEGRATION FOR THE
C**PULSE COMPRESSION APPROACH
C*****************************************************************
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMPLEX X(1),XINT
      NCOH=NFFT*NSAMPP
      DO 10 I=1,NCOH
      XINT=CMPLX(0.0,0.0)
      DO 11 INT=1,NCINT
      J=(NSTART-1)+I+(INT-1)*NCOH
      XINT=XINT+X(J)
   11 CONTINUE
      X(I)=XINT
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MTI(X,NIN,NPER,NPULSE,NOUT)
C*****************************************************************************
C**IMPLEMENTS THE DELAY LINE CANCELLER
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C**NIN IS THE NUMBER OF INPUT SAMPLES
C**NPULSE IS NUMBER OF PULSE CANCELLED
C**NPER IS NUMBER OF CODE PERIODS IN DELAY
C**NOUT IS NUMBER OF OUTPUT SAMPLES
C*****************************************************************************
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMPLEX XSTORE(150),YSTORE(150),X(1)
      MIN=NIN
      NDELAY=NSAMPP*NPER
      DO 15 IFILT=1,NPULSE
      MOUT=MIN+NDELAY
      DO 10 I=1,NDELAY
      XSTORE(I)=X(I)
   10 CONTINUE
      NDEL1=NDELAY+1
      DO 11 I=NDEL1,MIN,NDELAY
      DO 12 J=1,NDELAY
      YSTORE(J)=X(I+J-1)
   12 CONTINUE
      DO 13 J=1,NDELAY
      X(I+J-1)=YSTORE(J)-XSTORE(J)
   13 CONTINUE
      DO 14 J=1,NDELAY
      XSTORE(J)=YSTORE(J)
   14 CONTINUE
   11 CONTINUE
      MINP=MIN+1
      DO 16 J=MINP,MOUT
      X(J)=-YSTORE(J-MIN)
   16 CONTINUE
      MIN=MOUT
   15 CONTINUE
      NOUT=MOUT
      RETURN
      END
```

```
SUBROUTINE NOISE      74/74   OPT=1                    FTN 4.2+74355      1

      SUBROUTINE NOISE(X)
C*****************************************************************
C**ADDS NOISE TO THE INPUT SIGNAL
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C*****************************************************************
      COMMON/XNOISE/XMEAN,STDEV
      COMMON/PROPAR/NLOOK,NSAMPR,NFFT,NSCLK,NCINT
      DIMENSION XU(24)
      COMPLEX X(1),GAUSS
      DO 10 I=1,NLOOK
      CALL RANDU(XU,24)
      GAUSS1=0.0
      GAUSS2=0.0
      DO 11 J=1,12
      GAUSS1=GAUSS1+XU(J)
   11 GAUSS2=GAUSS2+XU(J+12)
      GAUSS1=(GAUSS1-6.0)*STDEV+XMEAN
      GAUSS2=(GAUSS2-6.0)*STDEV+XMEAN
      GAUSS=CMPLX(GAUSS1,GAUSS2)
   10 X(I)=X(I)+GAUSS
      RETURN
      END
```

```
      SUBROUTINE PLOT1P(X,MX,MSKIPX,MP,MSIZE,FRM,KPERIO)
C*****************************************************************************
C**GENERATES A LINE PRINTER PLOT
C*****************************************************************************
      DIMENSION X(1),XL(110),W(6)
      DATA BLANK,ASTRIX,HOR,VERT/1H ,1H*,1H-,1HI/
      DATA W(1)/10H(1X,I5,2X,/
      DATA W(3)/3HA1,/
      DATA W(4)/9H1X,E12.6)/
C     PERIODIC PLOT
C     X=ARRAY TO BE PLOTTED
C     MX=STARTING ADDRESS
C     MSKIPX=DISTANCE BETWEEN POINTS
C     MP = NUMBER OF POINTS PLOTTED
C     MSIZE = WIDTH OF THE PLOT(LESS THAN 110)
C     FRM = "MSIZE"
C     KPERIO = PERIOD
      WRITE(6,1)
    1 FORMAT(1H1)
      IF(MSIZE.GT.110) MSIZE=110
      IF(MSIZE.GE.110)FRM=3H110
      W(2)=FRM
      MMX=MX+MP-1
      MA=MX
      IF(MA.LE.0) MA=MA+KPERIO
      XMIN=X(MA)
      XMAX=X(MA)
      DO 5 JJ=MX,MMX,MSKIPX
      J=JJ
      IF(J.LE.0) J=J+KPERIO
      IF(J.GT.KPERIO) J=J-KPERIO
      IF(X(J).GT.XMAX) XMAX=X(J)
      IF(X(J).LT.XMIN) XMIN=X(J)
    5 CONTINUE
      IF((XMAX-XMIN).LE.1.E-7)XMIN=XMIN-1.E-5
      KS=1
      IF(XMIN.GT.0.0.OR.XMAX.LT.0.0) KS=2
      DELX=(XMAX-XMIN)/FLOAT(MSIZE-1)
      GO TO (7,8),KS
    7 NAXIS=1.0-XMIN/DELX
    8 CONTINUE
      DO 13 JA=MX,MMX,MSKIPX
      JX=JA
      IF(JX.LE.0) JX=JX+KPERIO
      IF(JX.GT.KPERIO) JX=JX-KPERIO
      DO 14 J=1,MSIZE
   14 XL(J)=BLANK
      IF(KS.EQ.1) XL(NAXIS)=VERT
      IX=1.0+(X(JX)-XMIN)/DELX
      IF(IX.GT.MSIZE) IX=MSIZE
      IF(IX.LT.1) IX=1
      GO TO (21,22),KS
   21 IF(IX.LE.NAXIS) GO TO 23
      NAXISP=NAXIS+1
      DO 24 I=NAXISP,IX
   24 XL(I)=ASTRIX
   23 IF(IX.NE.NAXIS) GO TO 25
```

```
      XL(IX)=ASTRIX
      GO TO 26
   25 NAXISM=NAXIS-1
      DO 27 I=IX,NAXISM
   27 XL(I)=ASTRIX
      GO TO 26
   22 IF(XMIN.GT.0.) GO TO 30
      DO 31 I=IX,MSIZE
   31 XL(I)=ASTRIX
      GO TO 26
   30 DO 28 I=1,IX
   28 XL(I)=ASTRIX
   26 CONTINUE
      WRITE(6,W)JA,(XL(I),I=1,MSIZE),X(JX)
   13 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PNCODE
C*****************************************************************
C**GENERATION OF THE PSEUDO RANDOM SEQUENCE
C*****************************************************************
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMMON/XCODE/NSTAGE,INIT(10),FEEDBK(10),CODE(1024)
      DIMENSION RGSTR(10)
      INTEGER FEEDBK,CODE,RGSTR
      DO 10 I=1,NSTAGE
   10 RGSTR(I)=INIT(I)
      DO 1 ICODE=1,NSAMPP,NSCLK
      KBACK=0
      DO 2 I=1,NSTAGE
      KBACK=KBACK+FEEDBK(I)*RGSTR(I)
    2 CONTINUE
      KBACKH=KBACK/2
      ICODEP=ICODE+NSCLK-1
      DO 4 JCODE=ICODE,ICODEP
      CODE(JCODE)=RGSTR(NSTAGE)
    4 CONTINUE
      DO 3 J=2,NSTAGE
      I=NSTAGE+2-J
      RGSTR(I)=RGSTR(I-1)
    3 CONTINUE
      RGSTR(1)=KBACK-2*KBACKH
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUANT(X)
C*****************************************************************
C**QUANTIZATION == A/D CONVERSION
C*****************************************************************
      COMMON/ADPAR/DELT,QAD,VMAX
      COMMON/PROPAR/NLOOK,NSAMPR,NFFT,NSCLK,NCINT
      COMPLEX X(1)
      DO 10 I=1,NLOOK
      X(I)=X(I)/VMAX
      XA=REAL(X(I))
      XB=AIMAG(X(I))
      AX=ABS(XA)
      BX=ABS(XB)
      IAX=AX/QAD
      IBX=BX/QAD
      AY=QAD*(FLOAT(IAX)+0.5)
      BY=QAD*(FLOAT(IBX)+0.5)
      XI=SIGN(AY,XA)
      XQ=SIGN(BY,XB)
      X(I)=CMPLX(XI,XQ)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RANDG(X,N,XMEAN,STDEV)
C*****************************************************************************
C**GENERATES GAUSSIAN RANDOM NUMBERS
C**N IS THE NUMBER OF NOISE SAMPLES
C**X IS THE OUTPUT ARRAY
C**XMEAN IS THE MEAN OF THE NOISE SEQUENCE
C**STDEV IS THE STANDARD DEVIATION
C**OUTPUT IS AN INDEPENDENT SEQUENCE
C*****************************************************************************
      DIMENSION X(1),XU(12)
      DO 10 I=1,N
      CALL RANDU(XU,12)
      GAUSS=0.0
      DO 11 J=1,12
   11 GAUSS=GAUSS+XU(J)
      GAUSS=(GAUSS-6.0)*STDEV+XMEAN
   10 X(I)=GAUSS
      RETURN
      END
```

```
SUBROUTINE RANDU      74/74   OPT=1                    FTN 4.2+74355      1

      SUBROUTINE RANDU(X,N)
C*************************************************************************
C**GENERATES UNIFORMLY DISTRIBUTED
C**RANDOM NUMBERS
C**N IS THE NUMBER OF NOISE SAMPLES
C**X IS THE OUTPUT ARRAY
C**OUTPUT IS AN INDEPENDENT SEQUENCE
C*************************************************************************
      COMMON/RANDM/ARD,RND1,RS
      DIMENSION X(1)
      DO 3 I=1,N
      ARN=ARD**2+RS**2
      K=ARN/1000000000.
      ARD=(ARN-FLOAT(K)*1000000000.)/1000.
      IF(ARD)2,1,2
    1 ARD=1.0
    2 RS=RS+1
      RND=ARD*0.000001
    3 X(I)=RND
      RETURN
      END
```

```
SUBROUTINE RMS        74/74   OPT=1                    FTN 4.2+74355      1

            SUBROUTINE RMS(X,NSTART)
          C*************************************************************************
          C**COMPUTES THE RMS VALUE OF EACH SAMPLE OF
          C**THE COMPLEX ARRAY X
          C**RESULT IN REAL PART OF ARRAY X
          C*************************************************************************
            COMPLEX X(1)
            COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
            NCOH=NFFT*NSAMPP
            NFIN=NSTART+(NCOH*NCINT)-1
            DO 10 I=NSTART,NFIN
            SSQ=CABS(X(I))
            X(I)=CMPLX(SSQ,0.0)
         10 CONTINUE
            RETURN
            END
```

```
      SUBROUTINE SIGNAL(X)
C*****************************************************************************
C**GENERATES SAMPLES OF SYNTHETIC VIDEO
C**INCLUDES TARGET RETURN,CLUTTER AND NOISE
C**CLUTTER INCLUDES IN RANGE CLUTTER
C**OUT OF RANGE DISTRIBUTED CLUTTER
C**FIXED CLUTTER, ANTENNA SPILLOVER
C**AND  OTHER ISOLATED CLUTTER
C*****************************************************************************
      COMPLEX X(1)
      COMMON/SIMPAR/NMONTE,INOPT(7),IPRINT(10)
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMMON/SIGPAR/FDOP,NDELTA
      COMMON/CLTPAR/SCRI,SCRD,SCRF,SPIL,SIGMAF,XM2,NCDELA,XM20,SIGMA0
     * ,NCDEL0,SCR0
      DO 10 I=1,NLOOK
   10 X(I)=CMPLX(0.0,0.0)
      IF(INOPT(1).EQ.1)CALL TARGET(X)
      IF(INOPT(2).EQ.1)CALL NOISE(X)
      IF(INOPT(3).EQ.0)GO TO 11
      CALL CLUTT(X,SCRI,XM2,SIGMAF,NDELTA)
   11 IF(INOPT(4).EQ.0)GO TO 12
      DO 13 ICELL=1,NSAMPP,NSCLK
      IRANG=(ICELL-1)/NSCLK
      DENOM=1.0+0.15*FLOAT(IRANG)
      WEIGHT=(1.0/DENOM)**3
      SCRDR=SCRD*3.4/WEIGHT
      CALL CLUTT(X,SCRDR,XM2,SIGMAF,ICELL)
   13 CONTINUE
   12 IF(INOPT(5).EQ.0)GO TO 14
      CALL CLUTT(X,SCRF,-1.0,0.0,NCDELA)
   14 IF(INOPT(6).EQ.0)GO TO 15
      CALL CLUTT(X,SCR0,XM20,SIGMA0,NCDEL0)
   15 IF(INOPT(7).EQ.0)GO TO 16
      CALL CLUTT(X,SPIL,-1.0,0.0,0)
   16 CONTINUE
      RETURN
      END
```

```
SUBROUTINE SPCTRM     74/74   OPT=1                    FTN 4.2+74355      1

      SUBROUTINE SPCTRM(X,N,NPLOT,IOPT)
C*****************************************************************************
C**COMPUTES AND PLOTS THE SPECTRUM OF X
C**X IS THE COMPLEX INPUT ARRAY
C**N IS THE NUMBER OF SAMPLES IN X
C**IOPT = 1 FOR MAGNITUDE PLOT ONLY
C**IOPT = 2 FOR PHASE PLOT ONLY
C**IOPT = 3 FOR BOTH PLOTS
C*****************************************************************************
      COMPLEX X(1),Y(1024)
      DIMENSION PLOT(1024)
      DATA FRM/3H100/
      DO 10 I=1,N
   10 Y(I)=X(I)
      CALL DFT(Y,N,-1,N)
      PI=3.141592654
      PI2=PI/2.0
      NP2=NPLOT/2
      NP2P=NP2+1
      NP2M=NP2-1
      IF(IOPT.EQ.2)GO TO 15
      DO 20 I=1,NP2P
   20 PLOT(I+NP2M)=CABS(Y(I))
      DO 21 I=1,NP2M
   21 PLOT(I)=CABS(Y(N+I-NP2M))
      CALL PLOT1P(PLOT,1,1,NPLOT,100,FRM,NPLOT)
      IF(IOPT.EQ.1)RETURN
   15 CONTINUE
      DO 30 I=1,NP2P
      YR=REAL(Y(I))
      YI=AIMAG(Y(I))
      IF(ABS(YR).LT.1.0E-09)GO TO 31
      PLOT(I+NP2M)=ATAN2(YI,YR)
      GO TO 30
   31 PLOT(I+NP2M)=SIGN(PI2,YI)
   30 CONTINUE
      DO 40 I=1,NP2M
      YR=REAL(Y(N+I-NP2M))
      YI=AIMAG(Y(N+I-NP2M))
      IF(ABS(YR).LT.1.0E-09)GO TO 41
      PLOT(I)=ATAN2(YI,YR)
      GO TO 40
   41 PLOT(I)=SIGN(PI2,YI)
   40 CONTINUE
      CALL PLOT1P(PLOT,1,1,NPLOT,100,FRM,NPLOT)
      RETURN
      END
```

```
      SUBROUTINE START
C**********************************************************************
C**INITIALIZATION SUBROUTINE
C**ALL THE PROGRAM INPUTS ARE READ IN NAMELIST
C**ARD,RND1,RS ARE INITIALIZATION PARAMETERS FOR
C**THE NOISE GENERATORS
C**NBIT IS THE WORD LENGTH OF A/D CONVERTER
C**FSAMP IS THE SAMPLING FREQUENCY
C**NSCLK = 1 IN THIS SIMULATION
C**VMAX  IS HALF THE DYNAMIC RANGE OF A/D CONVERTER
C**SNRDB IS SIGNAL=TO=NOISE RATIO IN DB
C**NSEC IS NO. OF SECOND ORDER SECTIONS IN THE
C**RECURSIVE NOTCH/LP FILTER
C**NFIRST IS 1 IF ORDER OF FILTER IS ODD
C**OTHERWISE NFIRST IS ZERO
C**NTYPE TYPE OF EACH SECOND ORDER AND FIRST
C**ORDER SECTION  LOGICAL ARRAY
C**.TRUE. FOR LP AND .FALSE. FOR HP FILTER
C**NOTE==ONLY BUTTERWORTH AND CHEBYSHEV FILTERS
C**CAN BE IMPLEMENTED IN THIS SIMULATION
C**FDOP IS THE DOPPLER FREQUENCY OF TARGET
C**NDELTA IS THE TARGET RANGE BIN NUMBER
C**NSTAGE IS NO. OF STAGES IN THE PN SEQUENCE GENERATOR
C**INIT IS THE INITIAL CONTENTS OF SHIFT REGISTER
C**FEEDBK IS THE FEEDBACK CONNECTION FOR THE
C**SHIFT REGISTER GENERATOR
C**NFFT IS THE NUMBER OF COHERENT INTEGRATION SAMPLES
C**NCINT = 1 IN THIS SIMULATION
C**NMONTE IS NOT USED IN THIS SIMULATION
C**INOPT IS THE INPUT OPTION ARRAY
C**INOPT(1)=1 = TARGET IS PRESENT
C**INOPT(2)=1 = NOISE IS PRESENT
C**INOPT(3)=1 = IN RANGE CLUTTER PRESENT
C**INOPT(4)=1 =DISTRIBUTED CLUTTER IS PRESENT
C**INOPT(5)=1 = FIXED CLUTTER IS PRESENT
C**INOPT(6)=1 = OTHER CLUTTER IS PRESENT
C**INOPT(7)=1 =ANTENNA SPILLOVER IS PRESENT
C**IPRINT IS NOT USED IN THIS SIMULATION
C**"""SCR STANDS FOR SIGNAL TO CLUTTER RATIO"""
C**SCRDBI IS SCR FOR IN RANGE CLUTTER IN DB
C**SCRDBD IS SCR FOR DISTRIBUTED CLUTTER IN DB
C**SCRDBF IS SCR FOR FIXED CLUTTER IN DB
C**SCRDBO IS SCR FOR OTHER CLUTTER IN DB
C**SPILDB IS ANTENNA SPILLOVER IN DB
C**NCDELA IS RANGE BIN OF FIXED CLUTTER
C**SIGMAF IS THE CLUTTER SPREAD FOR IN RANGE
C**AND DISTRIBUTED CLUTTERS
C**XM2 IS THE CLUTTER DC TO AC POWER RATIO FOR
C**IN RANGE AND DISTRIBUTED CLUTTER
C**NCDELO IS RANGE BIN OF OTHER CLUTTER
C**SIGMAO IS SPREAD OF OTHER CLUTTER
C**XM2O IS DC TO AC POWER RATIO OF OTHER CLUTTER
C**********************************************************************
      COMMON/ADPAR/DELT,QAD,VMAX
      COMMON/RANDM/ARD,RND1,RS
      COMMON/XNOISE/XMEAN,STDEV
      COMMON/FLTT/NSEC,NFIRST,NTYPE(10),CSEC(2,10),CFIRST,CONST
```

```
SUBROUTINE START      74/74   OPT=1                    FTN 4.2+74355      1

      COMMON/xCODE/NSTAGE,INIT(10),FEEDBK(10),CODE(1024)
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMMON/SIGPAR/FDOP,NDELTA
      COMMON/SIMPAR/NMONTE,INOPT(7),IPRINT(10)
      COMMON/CLTPAR/SCRI,SCRD,SCRF,SPIL,SIGMAF,XM2,NCDELA,XM20,SIGMA0
     * ,NCDEL0,SCR0
      INTEGER FEEDBK,CODE
      LOGICAL NTYPE
      NAMELIST/RNDGEN/ARD,RND1,RS
      NAMELIST/AD/NBIT,FSAMP,NSCLK,VMAX
      NAMELIST/NOIS/SNRDB
      NAMELIST/FLT/NSEC,NFIRST,NTYPE,CSEC,CFIRST,CONST
      NAMELIST/SIG/FDOP,NDELTA
      NAMELIST/PNC/NSTAGE,INIT,FEEDBK
      NAMELIST/PRO/NFFT,NCINT
      NAMELIST/SIM/NMONTE,INOPT,IPRINT
      NAMELIST/CLT/SCRDBI,SCRDBD,SCRDBF,SCRDB0,SPILDB,NCDELA,SIGMAF,
     * XM2,NCDEL0,SIGMA0,XM20
      NAMELIST/PRINT/NBIT,FSAMP,NSCLK,VMAX,SNRDB,FDOP,NDELTA,NSTAGE,
     1 INIT,FEEDBK,NFFT,NCINT,NMONTE,INOPT,IPRINT,SCRDBI,SCRDBD,
     2 SCRDBF,SCRDB0,SPILDB,NCDELA,SIGMAF,XM2,NCDEL0,SIGMA0,XM20
     3,NSEC,NFIRST,NTYPE,CSEC,CFIRST,CONST,ARD,RND1,RS
      READ(5,AD)
      READ(5,RNDGEN)
      READ(5,NOIS)
      READ(5,FLT)
      READ(5,SIG)
      READ(5,PNC)
      READ(5,PRO)
      READ(5,SIM)
      READ(5,CLT)
      WRITE(6,PRINT)
      DELT=1.0/FSAMP
      QAD=1.0/(2**(NBIT-1))
      SCRI=10**(0.1*SCRDBI)
      SCRD=10**(0.1*SCRDBD)
      SCRF=10**(0.1*SCRDBF)
      SCR0=10**(0.1*SCRDB0)
      SPIL=10**(0.1*SPILDB)
      SNR=10**(0.1*SNRDB)
      VAR=1.0/SNR
      STDEV=SQRT(VAR/2.0)
      XMEAN=0.0
      LENGTH=(2**NSTAGE)-1
      NSAMPP=LENGTH*NSCLK
      NCYCLE=NFFT*NCINT+5
      NLOOK=NCYCLE*NSAMPP
      CALL PNCODE
      DO 10 I=1,NSAMPP
   10 CODE(I)=2*CODE(I)-1
      RETURN
      END
```

```
SUBROUTINE TARGET     74/74   OPT=1                    FTN 4.2+74355

      SUBROUTINE TARGET(X)
C*****************************************************************
C**ADDS SAMPLES OF TARGET RETURN TO INPUT ARRAY
C**X IS THE COMPLEX INPUT/OUTPUT ARRAY
C**TARGET IS ASSUMED TO BE NON FLUCTUATING
C**TARGET RETURN POWER IS UNITY
C*****************************************************************
      COMMON/ADPAR/DELT,QAD,VMAX
      COMMON/PROPAR/NLOOK,NSAMPP,NFFT,NSCLK,NCINT
      COMMON/XCODE/NSTAGE,INIT(10),FEEDBK(10),CODE(1024)
      COMMON/SIGPAR/FDOP,NDELTA
      COMPLEX X(1)
      DIMENSION XR(4)
      INTEGER CODE
      TPI=6.283185307
      CALL RANDU(XR,4)
      PSI=XR(2)*TPI
      OMDOP=FDOP*TPI
      TIME=0.0
      NCYCLE=NLOOK/NSAMPP
      DO 10 IC=1,NCYCLE
      DO 11 ID=1,NSAMPP
      J=(IC-1)*NSAMPP+ID
      ICODE=ID-NDELTA
      IF(ICODE.LE.0) ICODE=ICODE+NSAMPP
      ARG=TIME*OMDOP+PSI
      XI=FLOAT(CODE(ICODE))*COS(ARG)
      XQ=FLOAT(CODE(ICODE))*SIN(ARG)
      X(J)=X(J)+CMPLX(XI,XQ)
   11 TIME=TIME+DELT
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TAYWG(X,NLOOK)
C*****************************************************************
C**IMPLEMENTS TAYLOR WEIGHTING FOR ANALOG PROCESSOR
C**X IN THE COMPLEX INPUT/OUTPUT ARRAY
C**NLOOK IS NUMBER OF SAMPLES IN X
C*****************************************************************
      COMPLEX X(1)
      DIMENSION F(9)
      DATA F/.462719,.126816E-1,.302744E-2,-.178566E-2,.884107E-3,
     *-.382432E-3,.121447E-3,-.417574E-5,-.249574E-4/
      N=13500
      N2=N/2
      DELANG=6.283185307/N
      DO 10 K=1,NLOOK
      WEIGHT=1.0
      DO 20 M=1,9
      INC=M*(K-1-N2)
      ANG=DELANG*INC
   20 WEIGHT=WEIGHT+F(M)*COS(ANG)*2.0
   10 X(K)=X(K)*WEIGHT
      RETURN
      END
```

```
SUBROUTINE WEIGHT     74/74   OPT=1                    FTN 4.2+74355      1

      SUBROUTINE WEIGHT(X,N)
C*****************************************************************************
C**HAMMING WEIGHTING PRIOR TO FFT PROCESSING
C*****************************************************************************
      DIMENSION X(1)
      COMPLEX X
      N2=N/2
      PI=3.141592654
      DELANG=PI/FLOAT(N)
      ANG=-PI
      DO 10 I=1,N
      ANG=ANG+DELANG
      WIND=0.54+0.46*COS(ANG)
      X(I)=X(I)*WIND
   10 CONTINUE
      RETURN
      END
```